uc3m | Universidad Carlos III de Madrid

Doble grado Ingeniería Informática y Administración y Dirección de Empresas

2018-2019

Trabajo Fin de Grado

Ataques DDoS con IoT, Análisis y Prevención de Riesgos

Javier Bautista Rosell

Tutores

Lorena González Manzano

Covadonga Gijón

Colmenarejo, octubre 2019

RESUMEN

La tecnología del Internet de las cosas (IoT) continúa creciendo a diario, lo que en el ámbito de la ciberseguridad aumenta las preocupaciones sobre la implicación de estos dispositivos en ataques informáticos a empresas e instituciones. Entre ellos los ataques de denegación del servicio distribuido (DDoS), que utilizan el envío de una gran cantidad de peticiones a un servidor para saturarlo. Este trabajo tiene tres objetivos: desarrollar un software capaz de leer el tráfico entrante en una red y permitir analizarlo, determinar la capacidad de los dispositivos IoT de participar en un ataque DDoS y valorar los daños que pueden causar estos ataques para precisar si a las empresas les es recomendable contratar servicios anti-DDoS.

En primer lugar se desarrolló un software para leer el tráfico de datos entrante en una red, tras lo cual se realizaron diversas pruebas con dos dispositivos IoT (Raspberry pi 3 y Node MCU) en un entorno controlado. Estas pruebas constituían diferentes ataques DoS contra un servidor en el que estaba instalado el software de lectura de datos. Los resultados de las pruebas mostraban un elevado potencial destructivo de ambos dispositivos en relación a su precio y tamaño, y que efectivamente estos dispositivos pueden suponer una amenaza si participan en ataques DDoS.

La valoración de los daños de los ataques DDoS a las empresas se realizó mediante el análisis de informes publicados por grandes empresas del sector de la ciberseguridad y otras instituciones relacionadas. Se concluyó que el coste de sufrir un ataque DDoS es muy elevado para todo tipo de empresas, superando el coste medio los 200.000$ por ataque, y que tanto el tamaño como la duración y la cantidad de los ataques aumenta año a año.

El crecimiento de los dispositivos IoT, junto con su capacidad para participar en ataques DDoS y el elevado daño causado por estos ataques presentan la necesidad de buscar soluciones contra estos ataques. Es necesario reducir la vulnerabilidad de los dispositivos IoT. Además, es recomendable que las empresas hagan frente a estas amenazas y contraten servicios anti-DDoS para protegerse de estos ataques cada vez más comunes y potentes.

Palabras clave:

Internet de las cosas, Ataques de denegación de servicio distribuido, Protección contra ataques DDoS.

ABSTRACT

The Internet of things (IoT) technology continues to grow daily, which raises concerns in the field of cybersecurity about the involvement of these devices in computer attacks on companies and institutions. Among these attacks are found the distributed denial of service attacks (DDoS), which send a huge amount of petitions to a server in order to saturate it. This investigation has three goals: to develop a software capable of reading incoming traffic on a network and allow analyzing it, to determine the capacity of IoT devices to participate in a DDoS attack, and to assess the damages that these attacks can cause to determine if it is advisable for companies to hire anti-DDoS services.

Firstly, a software was developed to enable the reading and analysis of the incoming traffic on a network. Then, several tests were done with two IoT devices (Raspberry pi 3 and NodeMCU), in a controlled environment. These tests were different DoS attacks against a server in which the developed software was installed. The results of these tests show a big destructive potential of both devices in relation to their price and size, and that they can consitute a considerable threat if they take part in a DDoS attack.

The damage assessment of DDoS attacks on companies was carried out by analyzing several reports published by large companies in the cybersecurity sector and other related institutions. The conclusion obtained is that the cost of suffering a DDoS attack is very high for all types of companies. In fact, the average cost reported exceeds $200.000 per attack. Furthermore, it is also inferred that both the size and duration and the number of DDoS attacks increase yearly.

The growth of IoT devices, together with their ability to participate in DDoS attacks and the high damage caused by these attacks present the need to find a solution to reduce the vulnerability of IoT devices to hackers. Furthermore, it is advisable that companies address these threats and hire anti-DDoS services to protect themselves from DDoS attacks.

Keywords:

Internet of things, Distributed denial of service Attacks, Protection against DDoS attacks.

Agradecimientos

A mi familia, en especial a mis padres, mi hermana y mi tía por aguantarme, apoyarme y ayudarme.

A mis compañeros de trabajo por el apoyo día a día y por preocuparse por el avance de este TFG.

A mi compañera de piso, que todavía no sé cómo no ha salido huyendo de la casa.

CONTENIDO

Índice de Figuras

Índice de Tablas

1. INTRODUCCIÓN

El Internet de las cosas (IoT) es una tecnología que va instalándose progresiva y definitivamente en el día a día de muchas personas de los países desarrollados. Para algunos de sus usuarios el término "IoT" es desconocido, pero no lo son los aspiradores Roomba, los televisores con conexión a Internet, las cámaras wi-fi, Alexa o Google home. Todos estos aparatos son dispositivos IoT y tienen en común su conexión a internet, pudiendo incluso recoger e intercambiar información sin necesidad de intervención humana. En definitiva, Internet ya no solo conecta personas, sino también cosas (Domodesk, 2014).

Los ataques de denegación de servicio, o DoS (Denial of Service), son ataques informáticos que pretenden inhabilitar la utilización de un sistema mediante la saturación del servidor que lo mantiene. Los servidores que alojan aplicaciones online tienen una determinada potencia y una capacidad limitada para procesar peticiones realizadas al mismo tiempo. Los ataques DoS se aprovechan de esta limitación enviando una gran cantidad de peticiones en un corto periodo de tiempo para colapsar el servidor (PaloAlto Networks, 2019).

Estos ataques informáticos no son nuevos, de hecho, el primer ataque DoS fue en el año 1974. Un tipo de ataques DoS desarrollado posteriormente son los ataques de denegación de servicio distribuido (DDoS). Estos ataques se comenzaron a utilizar alrededor del año 1995 y se caracterizan por utilizar varios dispositivos en lugar de uno para realizar el ataque, lo que puede generar ataques más potentes (Espinosa, 2017; Cox, 2014; Kaspersky, 2016).

Hay una relación entre los dispositivos IoT y los ataques DDoS. Pese a las legislaciones emergentes para securizar los dispositivos IoT, los fabricantes no suelen implantarles muchas medidas de seguridad. Esto facilita a los hackers el acceso a una multitud de pequeños dispositivos que pueden utilizar en su beneficio (Anscombe, 2019). Dado que la conexión a Internet de estos dispositivos les permite enviar peticiones a servidores, un hacker que tiene acceso a una red de dispositivos IoT puede utilizarlos para realizar ataques DDoS, lo que supone un problema no sólo para los dueños de los dispositivos controlados, sino también para los usuarios y los proveedores del servicio atacado.

Esta relación entre el IOT y el DDoS, el crecimiento imparable del uso de las tecnologías IOT y la creciente preocupación en el sector de la ciberseguridad fueron puntos clave que

motivaron esta investigación. Sin embargo, fue el descubrir que el ataque DDoS sufrido por la empresa DynDNS en octubre del 2016 y que afectó a servicios tan destacados como Netflix, Twitter y Spotify, que fue desarrollado utilizando cámaras wi-fi lo que finalmente impulsó a abordar este tema (Cáceres, 2016).

1.1. Objetivo

Este proyecto tiene tres objetivos. El primero consiste en desarrollar un sistema que permita leer el tráfico entrante en una red y analizarlo. El segundo, comprobar, gracias al sistema desarrollado, la potencia que tienen los dispositivos IoT para la realización de ataques DoS y DDoS. El tercer objetivo del trabajo es analizar el perjuicio que puede provocar a una empresa el ser víctima de un ataque DDoS y comprobar la rentabilidad de contratar servicios anti-DDoS.

Para llevar a cabo los dos primeros objetivos se ha realizado un desarrollo de software analizando las tecnologías disponibles para lograrlo y posteriormente se han utilizado dos dispositivos IoT para la realización de diferentes ataques contra el servidor en el que se ha instalado el software creado.

El tercer objetivo se ha llevado a cabo mediante la comparación de informes de distintas empresas e instituciones del sector de la ciberseguridad e interpretando los datos obtenidos.

1.2. Organización del documento

El documento se ha dividido en 11 secciones. Las tres primeras presentan la introducción al estudio y el estado de las investigaciones y las posteriores recogen el estudio realizado y sus conclusiones. En concreto, la sección 1 presenta el tema de estudio, las motivaciones para realizar el proyecto y sus objetivos. La sección 2 introduce términos necesarios para la comprensión del resto del documento. En la sección 3 se recoge el estado actual de las investigaciones de los temas tratados.

Las secciones 4,5,6,7 abordan el trabajo experimental desarrollado para obtener datos sobre la capacidad de los dispositivos IoT de realizar ataques DDoS: en las secciones 4 y 5 se abordan el diseño y el desarrollo del lector del tráfico entrante a la red; en la 6 y en la 7 se diseña el entorno de pruebas, se realizan ataques con los dispositivos IoT y se muestran los datos obtenidos.

Las secciones 8, 9 y 10 contienen los datos, la metodología y los resultados obtenidos de la investigación sobre el coste de recibir un ataque DDoS. Finalmente, en la sección 11 se exponen las conclusiones del proyecto, se proponen futuras vías de investigación en esta área y se presentan recomendaciones que empresas, particulares y gobiernos pueden seguir para reducir la amenaza que suponen los ataques DDoS.

2. CONCEPTOS PREVIOS

2.1. Ataques de denegación de servicio (DoS) y denegación de servicio distribuidos (DDoS)

Los ataques de denegación de servicio, DoS (Denial of Service) por sus siglas en inglés, son ataques cuyo objetivo es inhabilitar el uso de un sistema informático. Este tipo de ataques puede tener como objetivos servidores que mantienen, por ejemplo, webs o aplicaciones. Los ataques DoS también se realizan contra otros dispositivos o servicios, como el ataque realizado a la empresa DynDNS, una empresa proveedora de servicios de Sistema de Nombre de Dominio (DNS), en el año 2016 (Cox, 2014; Cáceres, 20016).

La base principal del funcionamiento de estos ataques consiste en enviar gran cantidad de peticiones de diferentes tipos a un mismo punto para que el servidor o la red a la que se envía no soporte la cantidad de paquetes recibidos y como consecuencia se produzca una interrupción del servicio proporcionado (ver figura 1).

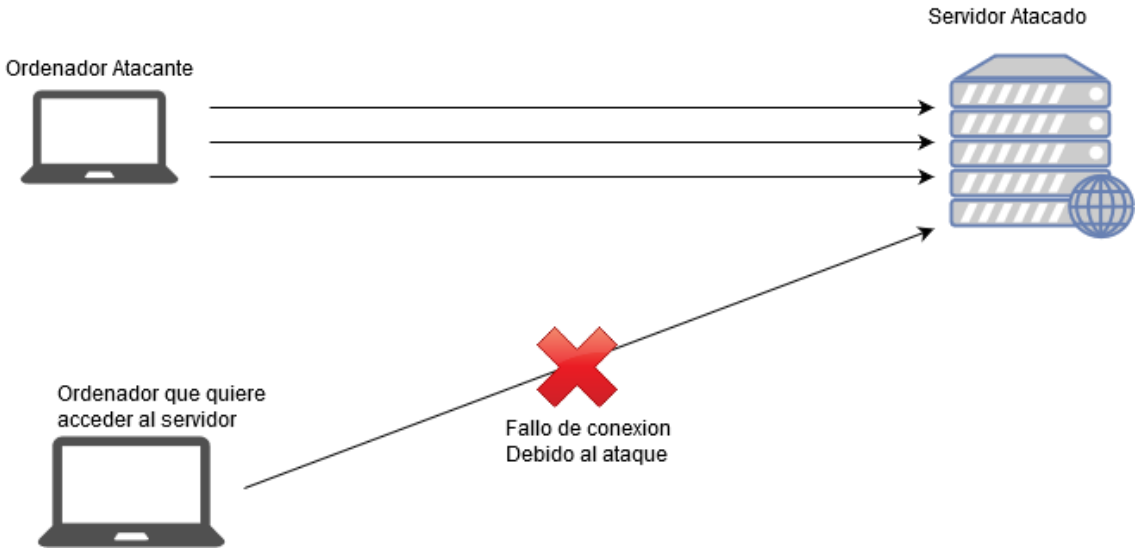

Figura 1 Ataque DoS a un Servidor web. Fuente: Elaboración propia

Los ataques de denegación de servicio distribuidos, DDoS (*Distributed Denial of Service*) por sus siglas en inglés, son un tipo de DoS en el cual el envío de peticiones está realizado por varios atacantes. Generalmente, estos ataques se realizan a través de botnets, que habitualmente están compuestas por ordenadores que han sido infectados y son controlados a distancia por los atacantes (Ver apartado 2.2).

2.1.1. Tipos de ataques DDoS

Existen diferentes tipos de ataques DDoS dependiendo del modo de ataque y del recurso atacado (Pandya, 2015).

2.1.1.1. Ataques DDoS según el modo de ataque

Hay dos tipos de ataques DDoS según el modo de ataque: el ataque directo y el ataque indirecto.

En el **ataque directo** las peticiones ilegítimas se envían directamente contra el *host* objetivo sin enmascarar las IPs atacantes (Pandya, 2015). (Ver figura 2).

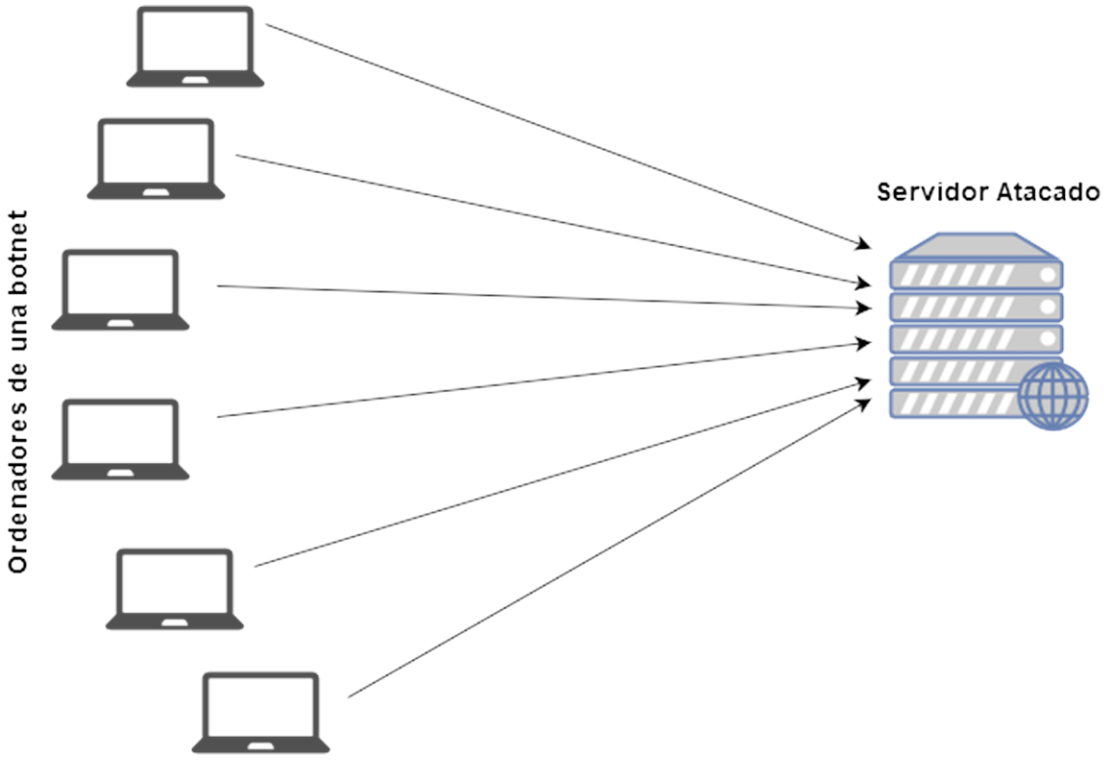

Figura 2 Ataque DDoS Directo. Fuente: Elaboración propia

El ataque indirecto es más complejo. En él se redistribuye el tráfico de las peticiones a través de intermediarios antes de atacar al *host* objetivo. De esta manera se ocultan las IPs atacantes, que son más difíciles de localizar. Además, el ataque indirecto permite llevar a cabo la técnica de amplificación, mediante la cual los propios dispositivos que actúan como intermediarios multiplican los paquetes, incrementando así la potencia del ataque (Pandya, 2015). (Ver Figura 3).

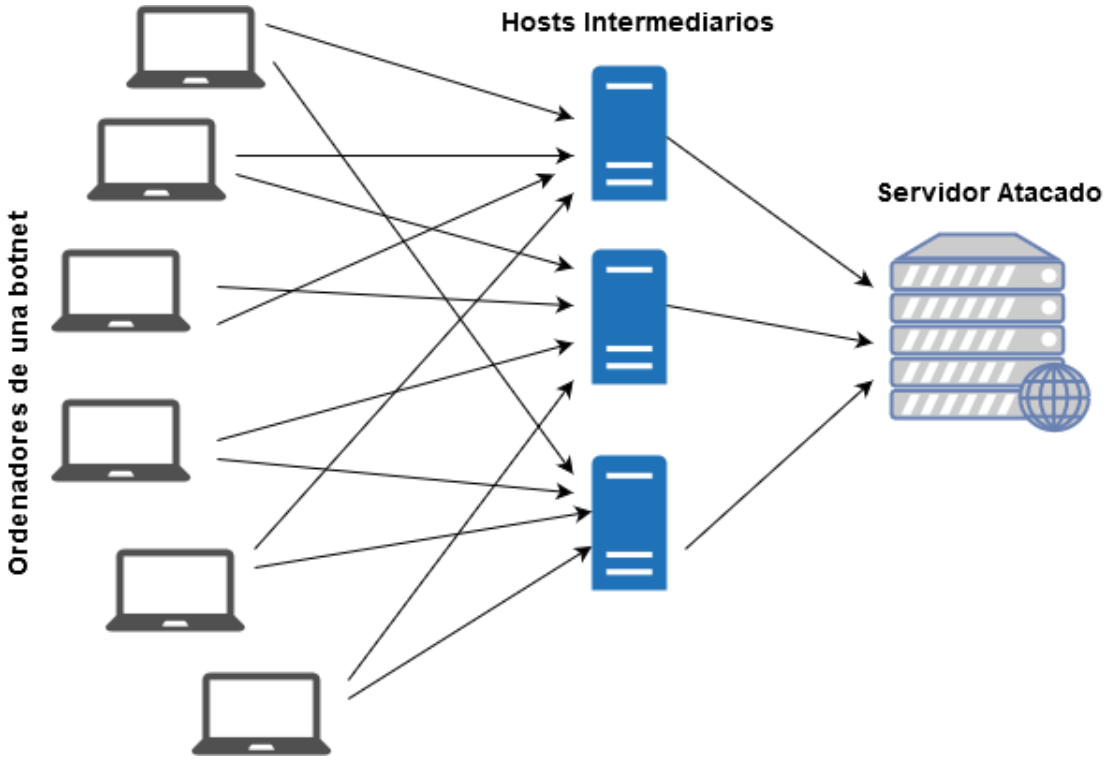

Figura 3 Ataque DDoS Indirecto. Fuente: Elaboración propia

2.1.1.2. Ataques DDoS según el recurso atacado

Los ataques DDoS pueden atacar diferentes recursos del host objetivo. Generalmente, se intenta saturar el ancho de banda de la red, la memoria o el uso del CPU (*central processing unit*) del sistema atacado.

Los ataques dirigidos al **ancho de banda de la red** tienen como objetivo consumir todo el ancho de banda de la red y no permitir al tráfico legítimo acceder a la red, provocando con ello la caída del servicio. Se pueden realizar de distintas maneras:

- UDP (*User Datagram Protocol*) Flood: Este tipo de ataque trata de "inundar" el host objetivo enviando una multitud de paquetes UDP a puertos aleatorios. Tal cantidad de peticiones satura el servidor (Cloudflare, 2019).
- ICMP (*Internet Control Message Protocol*) Flood: El planteamiento de este ataque es muy similar al anterior, con la excepción de que en lugar de paquetes UDP se utilizan paquetes ICMP, también llamados "ping" (Imperva, 2019a).
- LOIC (*Low Orbit Ion Cannon*): es una herramienta escrita en el lenguaje de programación C#, que permite realizar un ataque DoS sobre una IP o URL de

destino. Una vez configurada, esta herramienta genera un envío masivo de paquetes al destino establecido por el atacante (Motos, 2010).

El objetivo de los ataques dirigidos a la **memoria** es consumir la memoria del host objetivo provocando con ello su inhabilitación;

- Syn Flood: Este tipo de ataque se aprovecha del "the three-way-handshake", parte del protocolo de TCP (*Transmission Control Protocol*), que consiste en enviar una petición SYN al host, el cual responde a la petición con un ACK, que a su vez es contestado con otro ACK por el dispositivo que envió la petición inicialmente. Para la realización del ataque se mandan varias peticiones SYN, pero al recibir la respuesta del servidor atacado, no se envía la respuesta ACK que espera el SYN host. Esto provoca que el host quede a la espera de tantos ACK como SYN se han mandado, saturando la memoria (Jesus, 2010).
- Slowloris: Es muy parecido al anterior, pero en lugar de aprovecharse del protocolo "three-way-handshake" consume la memoria de su objetivo mediante peticiones HTTP (*Hypertext Transfer Protocol*) sin completar. Al recibir peticiones incompletas, el servidor se queda a la espera de que se completen las peticiones. Esto supone que, al acumularse muchas peticiones sin completar, se supere la capacidad de memoria del servidor, lo que impide que el tráfico legítimo pueda acceder (Systemadmin, 2009).
- HOIC (*High Orbit Ion Cannon*): es el predecesor del LOIC. En este caso el ataque se realiza enviando peticiones HTTP "GET" y "POST" al host objetivo hasta que este no pueda aguantar la cantidad de peticiones que tiene por resolver. (Imperva, 2019b)

Los ataques dirigidos a los **ciclos del CPU** se centran en consumir todo el uso del CPU del servidor atacado.

- Christmas tree: Este tipo de ataque consiste en enviar un paquete TCP específico aprovechando la configuración de los "*flags*" que se encuentran en la cabecera de los paquetes TCP. En este ataque se activan 3 *flags* específicos, "Urgent, Push, Fin" y, si el servidor no entiende el paquete, su CPU se consumirá hasta que termine por apagarse. Este tipo de ataque es muy conocido, por lo que es raro que pueda ser útil (Professor Messer, 2014).

2.1.1.3. Otras formas de clasificación

Basados en Volumen: en el que se encuentran UDP flood, ICMP flood, y LOIC-

Basados en protocolos: donde aparecen el christmas tree, y SYN flood

Ataques de capa de aplicación: comprenden el Slowloris y HOIC.

2.2. Botnets

De forma simplificada se puede definir una botnet como una red de ordenadores controlados por una misma persona de forma ilegítima. Pero realmente no tienen por qué ser solamente ordenadores, sino que también pueden ser móviles, neveras, cámaras, o cualquier elemento que tenga conexión a Internet.

La palabra botnet viene de las palabras inglesas "Robot" y "Network", debido a que los hosts controlados se pueden llamar robots, y a que todos ellos forman una red ("Network"). Otra forma de denominar a esos hosts controlados y la utilizada en este documento para referirse ellos es "Zombies". Una de las características más importantes de las botnets es su tamaño, pues cuantos más zombies tiene una botnet, más capacidad para realizar acciones malintencionadas posee (Fisher, 2013).

Para crear y expandir estas redes y conseguir el control de los hosts, un atacante puede utilizar un virus troyano, que es un tipo de virus que se oculta en programas aparentemente legítimos y que una vez ejecutado ese programa se instala en la máquina objetivo permitiendo controlarla (Fisher, 2013).

2.2.1. Usos de las botnets

Las botnets se pueden utilizar con diferentes propósitos, algunos de ellos son:

- Robo de información: el "Botmaster" (hacker que controla la botnet) utiliza la botnet para obtener la información sobre los usuarios de los zombies, pudiendo robar tanto datos personales como datos de empresas en caso de infectar ordenadores corporativos (Puri, 2003).
- Keylogger: otro de los usos es el robo de contraseñas. Esto puede suponer robo de datos bancarios para su posible uso, datos de acceso a redes sociales y todos los nombres de usuario y contraseñas que sean utilizados en el ordenador (Drupal, 2008).

- Envío de correos: estos zombies se pueden utilizar para envíos masivos de correos, que pueden ser spam, phishing, o bien pueden contener virus con el objetivo de incrementar el tamaño de la botnet (Drupal, 2008).
- Almacenamiento de datos: las botnets también se utilizan para almacenar datos ilegales en los zombies. De esta manera los dueños de los datos tienen acceso a ellos, pero no corren el riesgo de mantener esos datos en sus ordenadores. Por lo tanto, en el caso de que los datos sean descubiertos los que tendrían problemas serían los usuarios de las máquinas zombies, que no sabrían de la existencia de esos ficheros, y no el "Botmaster" que controla la botnet (Puri, 2003).
- Ataques DDoS: las botnets también pueden ser utilizadas para realizar peticiones masivas a servidores, con el objetivo de saturar un sistema (Puri, 2003). Este uso es el que desarrollamos en este trabajo.

2.2.2. Ataques DDoS con botnets

Las formas de controlar una botnet son variadas, ya que van evolucionando al mismo tiempo que la tecnología.

En el 2008 se empezaron a controlar las botnets mediante el IRC (*Internet Relay Chat*). El malware al instalarse abría un puerto TCP, se conectaba a un servidor IRC gestionado por el atacante y se mantenía a la escucha. El atacante, que controlaba la botnet, podía así manejar todos los ordenadores infectados mediante el servidor IRC (Puri, 2003).

Actualmente se utiliza el "Beaconing" que es más difícil de detectar. En esta forma de controlar las botnets el zombie está constantemente enviando "beacons", peticiones al servidor del atacante que no reciben respuesta, y que le indican que está a la escucha y que puede recibir órdenes. De esta manera, cuando el atacante quiera realizar un ataque DDoS solo tiene que mandar una respuesta especificando la ejecución de comandos deseados a los bots de su red.

El beaconing es difícil de detectar ya que no hay un patrón que identifique la presencia de los beacons en la red y que muchas veces la velocidad del ordenador tampoco indica su presencia. De acuerdo con la información de la web de Logrhythm (2016), uno de los líderes en el sector de los SIEMs (*Security Information and Event Management*), el beaconing puede ser realizado con cualquier frecuencia, desde una vez cada pocos segundos hasta una vez por semana o incluso con más diferencia de tiempo, lo que

dificulta su detección. Además, mientras otros virus ralentizan la velocidad del ordenador, en el caso de las botnets los ciberdelincuentes prestan atención a que esto no ocurra. Para ellos, es más beneficioso controlar muchas máquinas y consumir pocos recursos en cada una, que controlar pocas máquinas con muchos recursos.

Debido a estos problemas, los beacons no son fácilmente detectables incluso utilizando programas de análisis de red. Actualmente, una de las únicas maneras de asegurarse de que estos virus son descubiertos es tener un sistema SIEM, que controla y analiza todos los logs generados por las máquinas. Sin embargo, esta solución no es viable para los ordenadores personales debido a su alto coste.

¿Cómo afectan el uso de botnets a los ataques DDoS?

El uso de botnets facilita la realización de los ataques DDoS. En primer lugar, una vez que se tiene la botnet se puede realizar un ataque DDoS presionando un botón, y ya no es necesario contar con una multitud de personas que realicen peticiones al mismo tiempo. En segundo lugar, utilizar botnets facilita que una mayor cantidad de máquinas participen en el ataque, lo que permite disponer de una mayor potencia al realizar los ataques, incrementando la posibilidad de que el objetivo del ataque se cumpla. Finalmente, el uso de una botnet dificulta la detección y detención del ataque DDoS, al poder estar las IPs atacantes repartidas alrededor del mundo.

2.3. Internet de las cosas (IoT)

El Internet de las cosas (IoT, por sus siglas en inglés) se define como "un sistema de dispositivos de computación interrelacionados, máquinas mecánicas y digitales, objetos, animales o personas que tienen identificadores únicos y la capacidad de transferir datos a través de una red, sin requerir de interacciones humano a humano o humano a computadora" (Rouse, 2017).

Entre los dispositivos IoT actualmente se encuentran todo tipo de electrodomésticos, desde bombillas y cámaras, hasta neveras y televisores. Esto supone que en las casas de la mayor parte de la población de los países desarrollados haya este tipo de dispositivos.

El número de dispositivos IoT es cada vez mayor. En la figura 4 se muestra la previsión de dispositivos IoT frente a dispositivos no IoT desde 2015 hasta 2025. Se observa un alto crecimiento en los dispositivos IoT previendo incluso que en el 2022 lleguen a haber

más dispositivos IoT que dispositivos no IoT y casi duplicarlos en el 2025 (Instituto Nacional de Ciberseguridad [INCIBE], 2019).

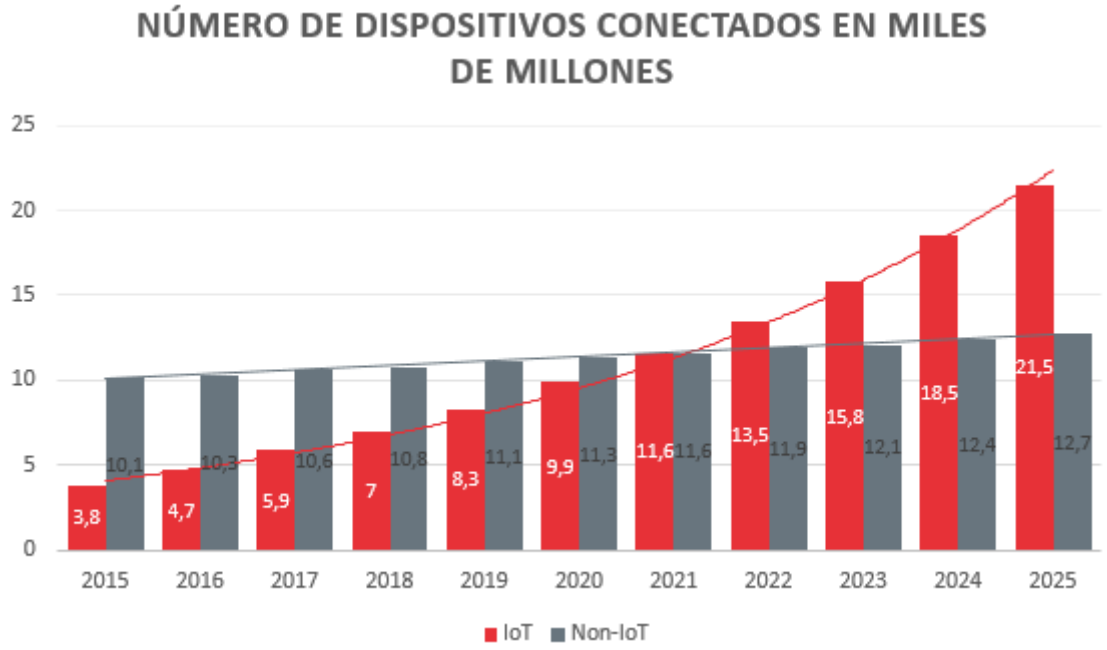

Figura 4 Evolución de los dispositivos IoT. Fuente: INCIBE (2019)

Este crecimiento por sí solo no es un problema, pero estos dispositivos pueden ser problemáticos si se consideran sus normalmente escasas medidas de seguridad informática. La poca seguridad con la que cuentan puede llevar a que algunos hackers se aprovechen de estos dispositivos para obtener datos sobre el dueño o controlarlos para realizar ataques DDoS. Algunos de los fallos de seguridad de este tipo de dispositivos son las contraseñas débiles, predecibles o dentro del código, los servicios de red inseguros, o la configuración poco segura por defecto, entre otros (Harán, 2019).

3. ESTADO DEL ARTE

3.1. Investigaciones previas sobre ataques DDoS con IoT

Existen multitud de noticias e investigaciones que relacionan los dispositivos IoT con ataques DDoS, pero el número de investigaciones que evalúan el potencial de los dispositivos IoT para efectuar los ataques DDoS es muy reducido.

Las noticias indican tres grandes ataques supuestamente llevados a cabo por dispositivos IoT. Estos ataques son los que recibieron "KrebsOnSecurity.com", "OVH" y "Dyn-DNS" entre septiembre y octubre de 2016. Según la Agencia de la Unión Europea para la ciberseguridad (ENISA) el ataque a OVH fue llevado a cabo por una botnet de más de 145000 dispositivos IoT entre cámaras IP y DVRs (ENISA, 2016).

Además de las noticias, que indican la participación de los dispositivos IoT en estos ataques, hay también investigaciones que relacionan el crecimiento de los dispositivos IoT con el aumento del número de ataques DDoS. Este es el caso del informe "Analysis of the IoT Impact on volume of DDoS Attacks" que en su conclusión indica que entre los años 2013 y 2015 ha habido un gran incremento de los ataques por capa de transporte frente a los ataques por capa de aplicación debido al aumento de dispositivos IoT (Perakovic, Perisa y Cvitic, 2015).

Por último, existen algunos informes que valoran la potencia de dispositivos IoT. En la realización de este trabajo solo ha sido posible encontrar uno debido a que son escasos. En este informe se comprueba la capacidad de un dispositivo IoT, una televisión, de realizar una amplificación de ataque DDoS tanto por TCP, UDP e ICMP. La conclusión del informe es que con apenas 37.890 dispositivos se podría llegar a generar un ataque de 36,28Tbps, que es una potencia mucho mayor de la de cualquier ataque DDoS sucedido hasta la fecha (Hengst, 2016).

Debido a los pocos informes sobre la valoración de la potencia de los dispositivos IoT existentes tampoco se han valorado muchos de estos dispositivos. En este proyecto se valoran dos de los dispositivos IoT más extendidos, una Raspberry Pi 3 y una placa de desarrollo, en este caso se ha valorado con una Node MCU, pero podría haberse utilizado cualquier placa de desarrollo.

3.2. Investigaciones previas de daños de los ataques DDoS

Respecto a las investigaciones sobre los daños producidos por ataques DDoS existen diversos informes realizados por empresas especializadas en el sector como Kaspersky, Arbor Networks y Neustar. Estos informes recurren a encuestas con personas en el sector y a datos sobre ataques realizados previamente para inferir los daños económicos y el riesgo reputacional que ocasiona sufrir un ataque DDoS (Kaspersky, 2014; Neustar, 2019a; Netscout y Arbor Networks, 2018).

El informe de Neustar (2019a) publicado en enero del 2019 realiza un sondeo entre 300 profesionales de ciberseguridad que posiciona los ataques DDoS como la tercera mayor amenaza de ataques informáticos. Ese mismo informe también indica que la protección contra ataques DDoS es el segundo tipo de seguridad informática que más aumentó en las empresas en 2018. Cabe destacar que, en un segundo informe publicado en abril de 2019, que también encuesta a profesionales del sector de la ciberseguridad, los ataques DDoS fueron considerados no la tercera, sino la mayor amenaza informática a la que se enfrentan las empresas actualmente (Neustar, 2019b).

Otros informes se centran en los costes económicos que este tipo de ataques causan a las empresas. Ponemon Institute (2012) encuesta a 705 profesionales informáticos e indica que el coste medio de un ataque DDoS es de 22.000 $ el minuto, pudiendo variar desde 1$ el minuto hasta 1.000.000$ el minuto según el tamaño de la empresa y la magnitud del ataque, siendo la duración media de dichos ataques 54 minutos.

Un tercer grupo de informes considera la cantidad de empresas sufriendo estos ataques y el coste reputacional de los mismos. Kaspersky (2014) realiza un sondeo entre 3900 representantes de empresas de 27 países, e indica que un 38% de las empresas encuestadas creen que un ataque DDoS daña a su reputación y que un 38% de los negocios de servicios financieros o con servicio de cara al público sufrieron ataques DDoS en el año del informe (Kaspersky, 2014).

Finalmente, otro tipo de informes tratan de inferir la actividad y el número de este tipo de ataques en un futuro. Los informes públicos sobre este tema son escasos y muy poco actualizados (e.g: Moore, Voelker y Savage, 2001).

Este trabajo compara distintos informes económicos para poder realizar un análisis de riesgos y convencer a las empresas con datos reales de si es necesario o no contratar sistemas anti-DDoS.

4. ANÁLISIS DEL SOFTWARE

4.1. Perspectiva general del sistema

Para desarrollar esta investigación se ha desarrollado un sistema cuyo objetivo es facilitar el análisis del tráfico entrante a un servidor mediante la lectura de los paquetes de datos entrantes en una determinada red y su visualización a través de gráficos. El uso de este sistema debe permitir al usuario conocer el tráfico habitual en su red y detectar un ataque DDoS. Para ello, el sistema deberá captar los paquetes entrantes y dividir la información contenida en estos paquetes según su protocolo de red con el fin de comprender su contenido. Esta división permitirá conocer el puerto atacado, la IP atacante, y el tamaño del paquete.

Para cumplir ese objetivo lo primero que se necesita es que el programa pueda diferenciar entre los tipos de protocolos ICMP, TCP y UDP. Se han elegido estos protocolos debido a que son los más conocidos y utilizados en ataques DDoS. También es necesario poder visualizar los datos de forma correcta para que el usuario pueda obtener información sobre el tráfico recogido.

El programa también debe de ser capaz de distinguir los diferentes elementos de los segmentos de los protocolos. Los elementos que deben analizarse varían entre los diferentes protocolos. El programa leerá y distinguirá los elementos pertinentes para facilitar el análisis de los paquetes recibidos.

Por último, dado que es una aplicación que se debe estar ejecutando 24/7, se necesita aplicar un control de logs que permita detectar y resolver los errores que puedan ocurrir durante su funcionamiento, como pueden ser los errores al cargar los datos a la base de datos BBDD.

4.2. Arquitectura del sistema

El sistema deberá ser fácilmente escalable con respecto a los tipos de paquetes y datos que puede analizar. Por esta razón, se ha decidido utilizar un diseño modular para el sistema en el que cada tipo de protocolo se encuentra en un módulo independiente. Asimismo, la base de datos del sistema y la aplicación encargada de la visualización de datos constituirán dos módulos adicionales.

La figura 5 muestra la arquitectura del sistema. La parte izquierda presenta la zona en la que los usuarios accederían a la información. En ella se encuentra el módulo de

visualización de información. En la parte derecha se encuentran los módulos que pertenecen al servidor, entre los que se hallan los tres capturadores de tráfico y la base de datos.

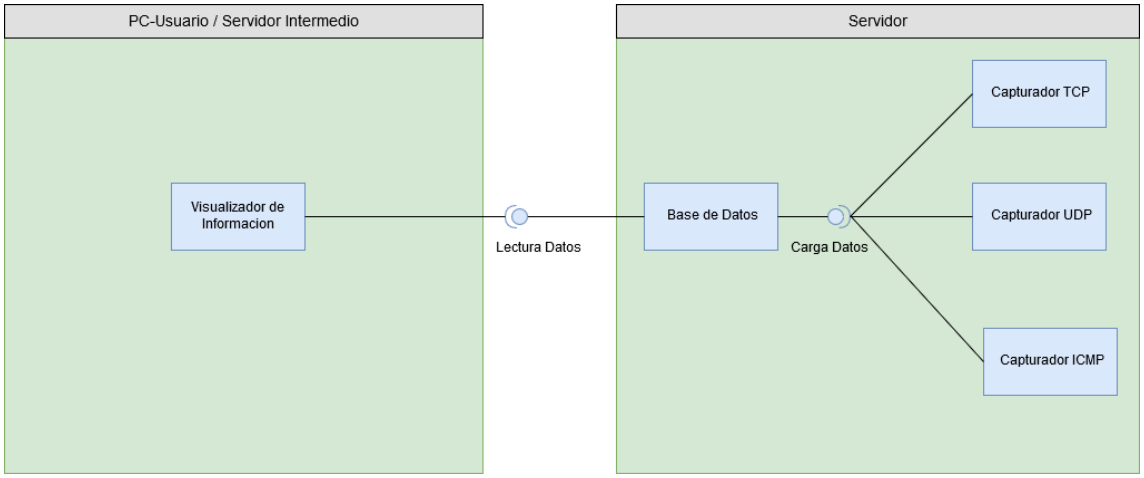

Figura 5 Arquitectura del sistema. Fuente: Elaboración propia

4.3. Estudio tecnológico

En este apartado se realiza un estudio de las posibles tecnologías que se pueden usar para realizar el desarrollo de los distintos módulos. Dado que en el desarrollo no se utilizan programas externos hasta la visualización de los resultados de las pruebas, no existe una tecnología impuesta para el desarrollo del programa.

4.3.1. Tecnologías para la base de datos

Para el desarrollo de la base de datos, se debe determinar la forma de almacenar los datos, para lo que existen múltiples alternativas. En este estudio se ha decidido examinar tres opciones, Mongo DB, MySQL y Maria DB, ya que MongoDB se encuentra entre las bases de datos más utilizadas actualmente y MySQL y MariaDB son las bases de datos relacionales más importantes (MongoDB, 2019; MySQL, 2019; MariaDB, 2019; pandorafms, 2015).

MongoDB es un sistema de bases de datos no relacionales y no SQL (*Structured Query Language*). En él se guardan los datos como documentos estilo JSON (*JavaScript Object Notation*). Sus principales ventajas son su velocidad de recuperación de información y que es gratuita. Entre las desventajas de estas bases de datos es la posibilidad existente de

inconsistencia al permitir cargar JSONs con un formato diferente al inicial (pandorafms, 2019).

MySQL sigue un sistema de bases de datos relacional. Su ventaja principal frente a MongoDB es que utiliza un sistema relacional, que además de permitir la recuperación de datos, permite relacionar distintas filas y columnas para generar información y asegura la consistencia de los datos marcando el tipo de datos de cada una de las columnas. Su principal desventaja es que recientemente se ha convertido en un sistema de gestión de bases de datos de pago.

MariaDB es una bifurcación de MySQL que apareció tras la adquisición de MySQL por Oracle. Esta base de datos tiene todas las ventajas de MySQL, además de ser gratuita.

Debido a que MariaDB no cuenta con las desventajas de MongoDB y MYSQL, y que para el proyecto desarrollado no es necesaria una rápida recuperación de los datos, se ha decidido hacer uso de este sistema de gestión de bases de datos para el desarrollo de la base de datos.

4.3.2. Tecnologías para la monitorización del tráfico

Para desarrollar el programa de monitorización del tráfico no se ha encontrado la forma de utilizar programas ya existentes utilizados para leer el tráfico de red como Wireshark. Debido a ello, fue necesario hacer uso de la programación. Para este tipo de programación se planteó el uso de tres posibles lenguajes, Python, Java, y C. Los dos primeros por ser de los programas de desarrollo más usados hasta noviembre de 2018 y el tercero por ser el que más documentación tiene sobre sockets —"Un socket, es un método para la comunicación entre un programa del cliente y un programa del servidor en una red." (masadelante.com, 2019) — (Python, 2019; The Cyber Security Hub, 2019).

Con el fin de comparar y escoger el programa de desarrollo se utilizaron tres criterios: la velocidad, la simplicidad de codificación y las bibliotecas de funciones incorporadas disponibles. La mayor ventaja de Java y de C es que, comparados con Python, ejecutan ciertos códigos con mayor rapidez a consecuencia de la forma de compilar los programas que tiene Python. Sin embargo, Python es más sencillo de codificar y tiene una amplia biblioteca de funciones incorporadas y librerias, lo que facilitará la implementación de otras funciones, si se necesita. Debido a estas ventajas se decidió utilizar Python para la programación de la monitorización del tráfico (EDUCBA, 2019; Goebelbecker, 2018).

4.3.3. Tecnologías para la visualización

Por último, es necesario tomar la decisión sobre qué tecnología utilizar para la visualización de datos. Se ha decidido examinar Thinger.io y QlikView, ya que son dos programas que se han utilizado anteriormente para la visualización de tipos de datos similares con resultados satisfactorios (Thinger.io, 2019; Qlik, 2019).

La principal ventaja de thinger.io es la capacidad de visualizar los datos a tiempo real. Sin embargo, requiere conectar el servidor con la plataforma, lo que supone una desventaja ya que incrementa su dificultad de uso. Esta dificultad es aún mayor si se considera la decisión de usar Python para la monitorización del tráfico (apartado 4.3.2), porque thinger.io está principalmente enfocado a dispositivos IoT, que normalmente son programados C o en C#.

Qlik tiene la desventaja de que la visualización no sucede en tiempo real. Sin embargo, proporciona una serie de ventajas que facilitan el desarrollo y uso del sistema.

En primer lugar, es compatible con MariaDB, la tecnología usada para la base de datos, y permite conectar y obtener los datos de la plataforma de una manera sencilla. En segundo lugar, una vez que los datos están cargados en la plataforma, Qlik permite generar multitud de gráficos y tablas diferentes combinando las columnas de la base de datos como el usuario prefiera. Incluso permite programar operaciones básicas como sumas y conteos. Por último, Qlik proporciona la posibilidad de aplicar filtros para que el usuario pueda obtener información más precisa.

Debido a las ventajas de Qlik, se ha decidido utilizar este programa para la visualización de datos.

4.4. Requisitos del software

En los siguientes apartados se muestran los requisitos funcionales y no funcionales del software, siguiendo la guía del IEEE830-1998 (Institute of Electrical and Electronics Engineers [IEEE], 1998).

4.4.1. Requisitos funcionales

En la tabla 1 se muestra la información de los requisitos funcionales de acuerdo con el formato definido en el anexo 4.

Tabla 1

Requisitos funcionales.

Requisitos del software				
Tipo: Funcional				
Id	Nombre	Descripción	Estabilidad	Prioridad
RF-01	Capturar paquetes TCP	Detectar y leer los paquetes TCP entrantes al servidor	Alta	Alta
RF-02	Capturar paquetes UDP	Detectar y leer los paquetes UDP entrantes al servidor	Alta	Alta
RF-03	Capturar paquetes ICMP	Detectar y leer los paquetes ICMP entrantes al servidor	Alta	Alta
RF-04	Identificar IP origen	Debe ser capaz de identificar la IP origen en el paquete	Alta	Alta
RF-05	Identificar puerto destino	Debe ser capaz de identificar el puerto destino en el paquete	Alta	Alta
RF-06	Identificar tamaño	Debe ser capaz de identificar el tamaño en bits el paquete	Alta	Alta
RF-07	Contar Paquetes	Debe ser capaz de contar el número de paquetes con la misma IP origen y el mismo puerto destino	Alta	Alta
RF-08	Base de Datos	Debe ser capaz de conectar con la base de datos y cargar los datos recogidos	Media	Alta
RF-09	Logs	Debe ser capaz de mantener un registro de logs en ficheros de texto	Alta	Alta
RF-10	Base de Datos Lectura	La base de datos debe ser accesible por el programa de visualización	Alta	Alta

Nota. Fuente: Elaboración propia

4.4.2. Requisitos no funcionales

La tabla 2 muestra la información de los requisitos no funcionales siguiendo el formato definido en el anexo 4.

Tabla 2

Requisitos no funcionales.

Requisitos del software				
Tipo: No Funcional				
Id	Nombre	Descripción	Estabilidad	Prioridad
RNF-01	Disponibilidad	Debe estar ejecutándose 24/7	Alta	Alta
RNF-02	Base de datos securizada	La base de datos solo puede ser accesible por el administrador y por el programa de visualización.	Alta	Alta

Nota. Fuente: Elaboración propia

4.5. Casos de uso

4.5.1. Diagrama de casos de uso

Se han identificado dos casos de uso para este proyecto. En el caso principal, el actor es el usuario de la aplicación y puede consultar los datos del tráfico de red a través de Qlik. En el segundo caso, el actor es el administrador de la aplicación y puede leer los ficheros de log. Ambos casos se muestran en la figura 6.

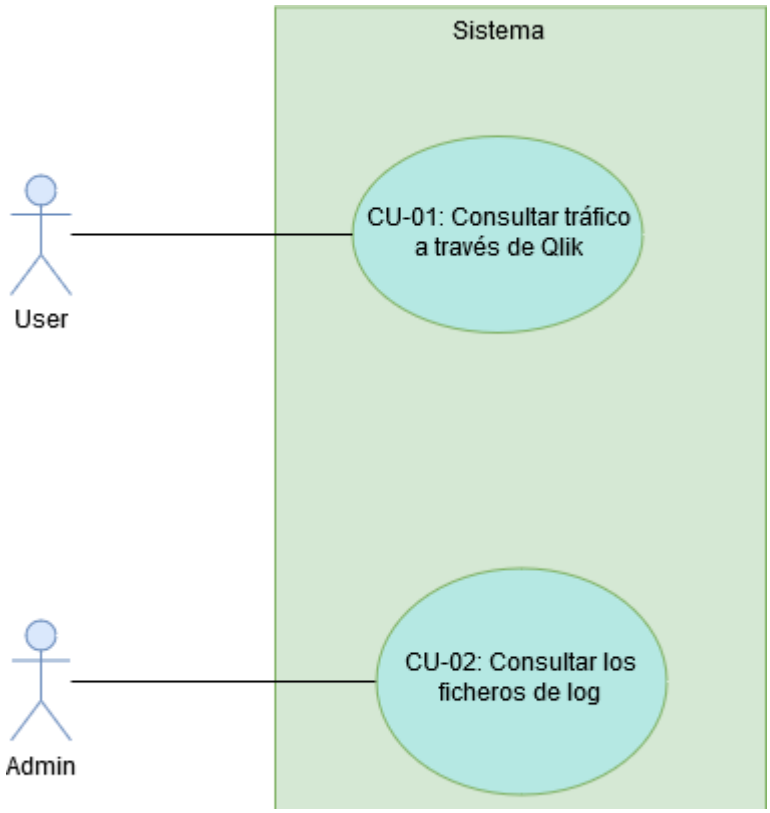

Figura 6 Diagrama de casos de uso. Fuente: Elaboración propia

4.5.2. Especificación de casos de uso de alto nivel

La tabla 3 especifica el primer caso de uso identificado. En este caso, un usuario puede visualizar los datos del tráfico a través de la aplicación Qlik.

Tabla 3

Caso de uso Consultar tráfico a través de Qlik.

Caso de uso	Consultar tráfico a través de Qlik
ID	CU-01
Actores	Usuario
Descripción	Permite la consulta de los datos de tráfico a partir de Qlik, filtrado de los datos, etc.
Precondiciones	1. La base de datos no está vacía
Postcondiciones	1. Los datos quedan cargados en Qlik 2. El usuario obtiene la información que le sea relevante

Nota. Fuente: Elaboración propia

La tabla 4 muestra el segundo caso de uso definido. Consiste en la consulta de los ficheros de log, para lo cual el administrador puede hacer uso de la aplicación de lectura de texto que prefiera.

Tabla 4

Caso de uso Consultar los ficheros de log.

Caso de uso	Consultar los ficheros de log
ID	CU-02
Actores	Administrador
Descripción	Permite la consulta de los ficheros de log por parte del administrador
Precondiciones	1. La captura de datos está activada 2. Existen los ficheros de log
Postcondiciones	1. El administrador puede comprobar el funcionamiento de la aplicación.

Nota. Fuente: Elaboración propia

4.6. Diagramas de secuencia

En este apartado se muestran y explican los diagramas de secuencia que corresponden a los dos casos de uso definidos en el apartado anterior.

Caso de uso principal (CU-01): consultar datos del tráfico a través de Qlik

En los diagramas de secuencia se han evitado todas las interacciones del trabajo de los datos dentro de Qlik que no tengan relación directa con la base de datos para simplificar los diagramas de secuencia.

En la figura 7 se muestra el diagrama de secuencia realizado en el caso de que el usuario necesite actualizar la base de datos antes de empezar a comprobar los datos.

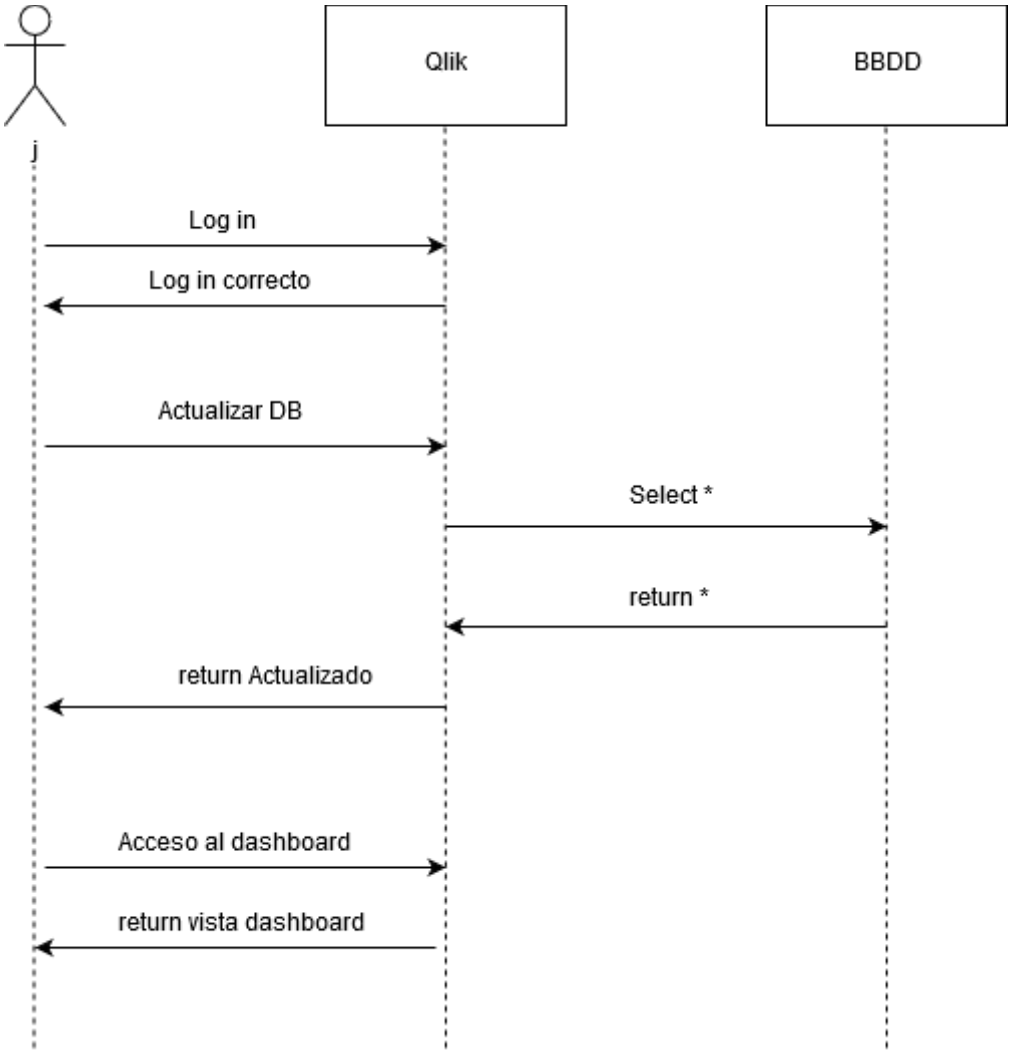

Figura 7 Diagrama de secuencia CU1. Fuente: Elaboración propia

En caso de tener ya los datos actualizados o de no necesitar actualizar los datos que tiene Qlik ya cargados, el diagrama de secuencia será el mostrado en la figura 8.

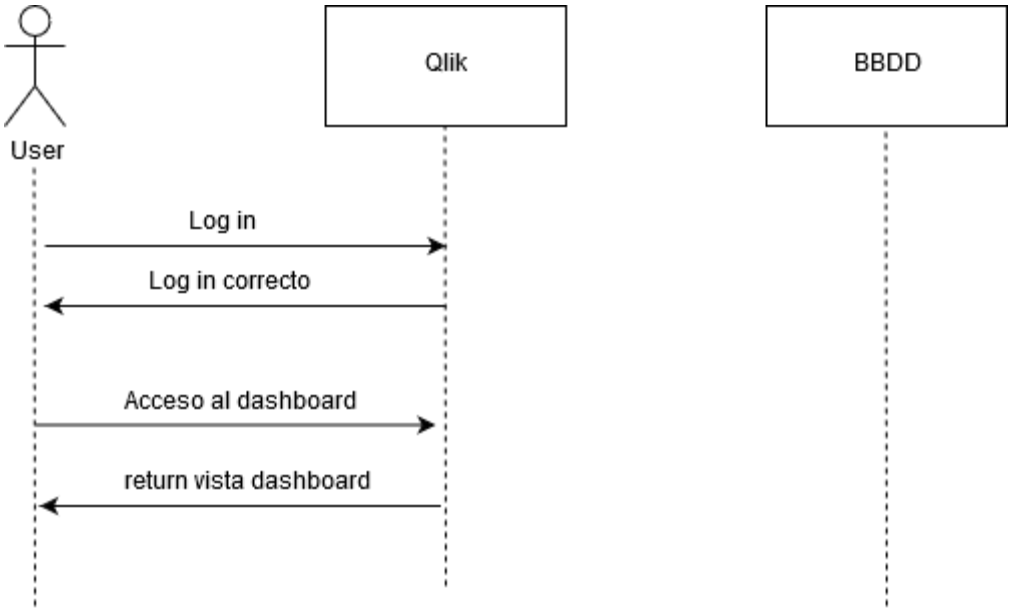

Figura 8 Diagrama de secuencia CU1 (2). Fuente: Elaboración propia

Caso de uso secundario (CU-02): consultar los ficheros de log

A los ficheros de log aun siendo parte de la aplicación no se accede a través de la base de datos ni de ninguno de los códigos, si no que el administrador accede directamente al servidor para comprobarlos como se muestra en el diagrama de la figura 9.

Para acceder a los ficheros de log no es necesario el uso de la base de datos o de ninguno de los códigos utilizados en el desarrollo, aunque sean parte de la aplicación. El administrador puede acceder a estos ficheros conectándose directamente al servidor, como se muestra en la figura 9.

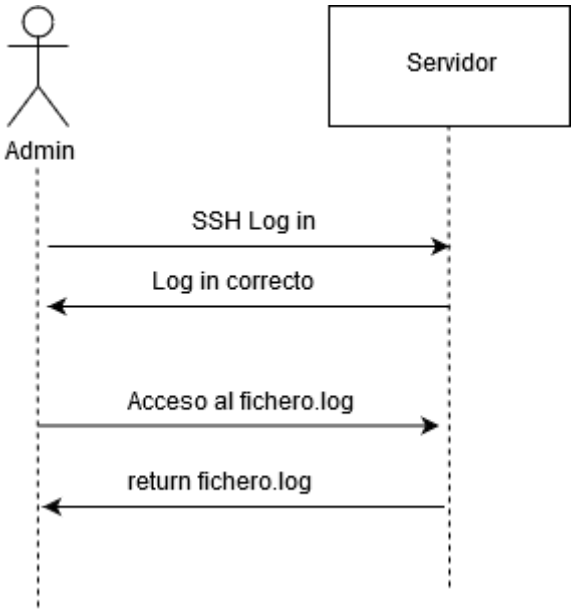

Figura 9 Diagrama de secuencia CU2. Fuente: Elaboración propia

4.7. Diseño del plan de pruebas de aceptación

Una vez definidos los requisitos y los casos de uso del código, es necesario realizar el plan de pruebas de aceptación de esos requisitos. En este apartado se detalla el plan de pruebas y muestra las verificaciones que se deberán realizar posteriormente para comprobar el correcto funcionamiento del programa.

Tabla 5

Prueba de aceptación 01.

ID prueba	PA-01
Descripción	Prueba de lectura de paquetes TCP
Requisitos a comprobar	RF-01, RF-04, RF-05, RF-06, RF-08
Precondición	Programa está activo
Acción	Enviar un paquete TCP
Postcondición	Se debe registrar el paquete en la base de datos

Nota. Fuente: Elaboración propia

Tabla 6

Prueba de aceptación 02.

ID prueba	PA-02
Descripción	Prueba de lectura de paquetes UDP
Requisitos a comprobar	RF-02, RF-04, RF-05, RF-06, RF-08
Precondición	Programa está activo
Acción	Enviar un paquete UDP
Postcondición	Se debe registrar el paquete en la base de datos

Nota. Fuente: Elaboración propia

Tabla 7

Prueba de aceptación 03.

ID prueba	PA-03
Descripción	Prueba de lectura de paquetes ICMP
Requisitos a comprobar	RF-03, RF-04, RF-05, RF-06, RF-08
Precondición	Programa está activo
Acción	Enviar un paquete ICMP
Postcondición	Se debe registrar el paquete en la base de datos

Nota. Fuente: Elaboración propia

Tabla 8

Prueba de aceptación 04.

ID prueba	PA-04
Descripción	Prueba de conteo de paquetes
Requisitos a comprobar	RF-06, RF-07, RF-08
Precondición	Programa está activo
Acción	Enviar varios paquetes iguales
Postcondición	Se deben registrar los paquetes, el número de ellos, y el tamaño en la base de datos

Nota. Fuente: Elaboración propia

Tabla 9

Prueba de aceptación 05.

ID prueba	PA-05
Descripción	Prueba de acceso a logs
Requisitos a comprobar	RF-09
Precondición	Programa ya ha realizado cargas en la base de datos
Acción	Entrar en el servidor y acceder a los ficheros de log
Postcondición	Se deben poder leer los ficheros de log con claridad

Nota. Fuente: Elaboración propia

Tabla 10

Prueba de aceptación 06.

ID prueba	PA-06
Descripción	Prueba de lectura de datos por Qlik
Requisitos a comprobar	RF-10
Precondición	Programa ya ha realizado cargas en la base de datos
Acción	Cargar datos en Qlik
Postcondición	Deben estar cargados los datos y poder ser visualizados

Nota. Fuente: Elaboración propia

5. DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

5.1. Diseño del software

En este apartado se detalla la descripción de los componentes, TrafficReader y base de datos, identificados en el análisis (sección 4). El tercer componente, de visualización de datos, no se describe en este apartado dado que es un componente externo (Qlik), y por lo tanto no se ha necesitado desarrollar para el funcionamiento del programa.

La base de datos es un componente clave para el sistema, porque es la comunicación que utiliza Qlik para obtener los datos. El código desarrollado se encarga de, una vez recolectados los datos, cargarlos a la base de datos para que puedan ser utilizados con posterioridad.

El componente desarrollado es el componente TrafficReader. Está compuesto de tres funciones, getTrafic, ChargeDatabase y resetValues. Es el componente más importante del sistema dado que se encarga de la lectura y el tratamiento del tráfico entrante al servidor.

5.2. Diseño de la base de datos

La base de datos, como se introdujo en el análisis (apartado 4.3.1), es MariaDB. MariaDB es una base de datos relacional. Para medir las variables necesarias, indicadas a continuación, y cumplir los requisitos expuestos en el apartado 4.4.1, se usará una tabla con seis columnas.

#	Nombre	Tipo de datos
1	IpOrigen	VARCHAR
2	PuertoDestino	VARCHAR
3	Bits	DOUBLE
4	NumeroPaqu...	DOUBLE
5	FechaYHora	DATETIME
6	Protocolo	VARCHAR

Figura 10 Diseño de la base de datos.
Fuente: Elaboración propia

La figura 10 presenta las variables (columnas) necesarias para controlar el tráfico habitual de la red y para detectar si un puerto está recibiendo un ataque de DDoS o una IP está atacando. Como está indicado en la imagen, se necesitará la IP del origen, el puerto de destino, el tamaño del paquete (los datos en bits), el número de paquetes recibidos, la fecha y hora en la que han sido recibidos y el protocolo. En caso de ataque, esta última variable (protocolo) permite detectar qué protocolo está siendo utilizado en el ataque. El tipo de datos que se usará para registrar cada entrada también se muestra en la imagen.

5.3. Diseño del código

El código, como se indicó en la "Arquitectura del sistema" (apartado 4.2), se divide en tres módulos independientes, uno por cada tipo de protocolo considerado, TPC, UDP e ICMP. Dado que las diferencias entre ellos son mínimas en la programación, este apartado explicará el diseño general del código, siendo éste aplicable a los tres módulos.

En el código se identifican tres funciones: dos principales y una secundaria. Entre las principales se encuentran *getTrafic*, la función de lectura de tráfico entrante y *chargeDatabase*, que se ocupa de la carga de los datos del tráfico identificado en la base de datos. La función secundaria, *resetValues,* se encarga del reinicio de las variables cada vez que se realiza una carga. La figura 11 representa un esquema de la clase genérica de TrafficReader.

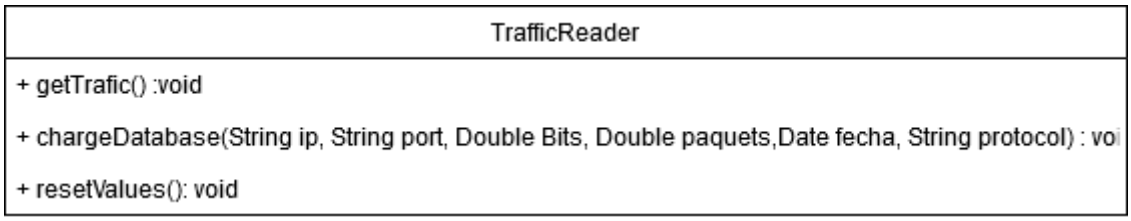

Figura 11 Esquema de clase de TrafficReader. Fuente: Elaboración propia

5.4. Código

En este apartado se detallan las decisiones tomadas para el desarrollo del código que se encarga de leer el tráfico entrante en el servidor y cargarlo en la base de datos. El desarrollo del código se llevó a cabo en dos etapas. Ya que Python fue el programa escogido para desarrollar el código (apartado 4.3.2), la primera etapa consistió en examinar como trabajaba este lenguaje con los paquetes de datos, y finalizó en la selección de un código base. Este código base fue modificado en la segunda etapa con el fin de seleccionar solamente los campos que se han previsto como necesarios y realizar la carga en la base de datos.

Etapa 1: Selección del código base

Para iniciar la primera etapa se investigó como trabajaba Python con los paquetes de red. Esta investigación concluyó con el descubrimiento de la librería "socket". Esta librería permite gestionar los sockets del servidor para poder leer los paquetes entrantes o enviar paquetes (Python, 2019).

Posteriormente se investigó sobre los códigos ya existentes en el ámbito de la captura de paquetes de datos para encontrar la posibilidad de reutilizar código. Todos los códigos hacían uso de la librería socket de la misma forma. Sin embargo, la mayoría de los códigos encontrados no traducían los paquetes de datos entrantes correctamente o se limitaban solo a contar el número de paquetes que se recibían. Finalmente **se escogió "Parsing the Sniffed packet" como código base,** ya que se consideró el código más completo y que permite leer, desempaquetar y traducir los paquetes de datos entrantes correctamente. Para ello, utiliza las librerías "socket", "sys" y "struct" (Silver Moon, 2011).

"Parsing the sniffed package" proporciona más datos de los considerados necesarios en el análisis del software sección 4 apartado 4.1. Sin embargo, estos datos podrían ser utilizados para futuras mejoras de la utilidad del software en actualizaciones futuras.

Etapa 2: Adaptación del código base

En su estado original "Parsing the sniffed packet" no permite cargar los datos recogidos en una base de datos ni genera ficheros de log que faciliten la detección y resolución de errores. Tampoco permite la lectura de paquetes UDP o ICMP, ya que está programado para solo leer paquetes TPC. Con el fin de resolver estos problemas y adaptar el código a los requisitos del sistema mencionados en el apartado 4.6, ha sido necesario modificar el código base.

Para evitar una carga excesiva de procesamiento se ha fijado que las cargas en la base de datos se realicen una vez por minuto.

Para conseguir que las cargas se realicen cada minuto sin perder datos, es necesario el uso de cuatro arrays (vectores). IPOrigen[], puertoDestino[], Bits[], numPaquetes[], y dos variables, "hora" y "horaProxima". La función de los arrays consiste en facilitar la identificación de cada tipo de paquetes. Cuando un paquete nuevo es leído por el código, este compara la IP origen del paquete con las IPsOrigen existentes en el array; en caso de existir pasa a comparar el puerto destino y, si también coinciden, entonces el tamaño del paquete se suma en esa posición del array Bits[] y se suma uno en el array numPaquetes[]; en caso contrario se añadirán los datos al final de cada uno de los arrays.

A su vez se ha determinado que, para simplificar, los logs se generarán a la vez que la carga. Si la carga se produce de forma exitosa, se registrará en el fichero del log "[*HoraActual*]-Carga: *Datos* ----" por cada fila cargada. En caso de que alguna de las cargas de filas dé error, en ese punto se escribirá en el fichero del log "[*HoraActual*]-Error de carga inesperado *Código de error*".

El código de la carga es la parte del código de la aplicación que ha sido completamente incluido y no basado en el código "Parsing the sniffed packet". El resultado de esta función al final del desarrollo es el mostrado en la figura 12. El resto de código y otros códigos utilizados se muestran en el anexo 3.

```
70  def Carga(IpSource, puerto, tamano, numero, hora):
71      f=open('logs/TCP.log', 'ab')
72      for count in range(0,idLista):
73          anadir="""Insert into Trafico values(%s, %s, %s, %s, %s, %s)"""
74          protocolo="TCP"
75          try:
76              datos=(IpSource[count], puerto[count], tamano[count], numero[count], hora, protocolo)
77              cursor.execute(anadir, datos)
78              mariadb_connection.commit()
79              f.write("["+hora.strftime("%H:%M:%S")+"]"+"Carga:  "+ str(datos) +"---- "+"\n\n")
80          except Exception as e:
81              f.write( "["+hora.strftime("%H:%M:%S")+"]"+ "error de carga inesperado:  " + str(e))
82      f.close()
83      reiniciarVariables()
84
```

Figura 12 Código de carga en la base de datos. Fuente: Elaboración propia

5.5. Resultados de las pruebas

En este apartado se muestran los resultados de las pruebas definidas (apartado 4.7). Para llevar a cabo las pruebas de envío de paquetes de manera controlada se ha utilizado el

programa Packet Sender (Packet Sender, 2019). Esta aplicación permite enviar paquetes con el protocolo y tamaño deseados al servidor que se quiere probar.

Como se puede ver en la tabla 11 se han superado todas las pruebas identificadas en el análisis.

Tabla 11

Resultados de las pruebas de aceptación.

ID Prueba	Resultado
PA-01	Superada
PA-02	Superada
PA-03	Superada
PA-04	Superada
PA-05	Superada
PA-06	Superada

Nota. Fuente: Elaboración propia

6. EVALUACIÓN

6.1. Laboratorio

Para la realización de un estudio sobre la potencial capacidad de los dispositivos IoT de participar activamente en ataques DDoS era necesario utilizar un entorno controlado para la ejecución de las pruebas. Por esta razón se decidió montar un laboratorio dentro de una red privada, de modo que los ataques se realizasen en este entorno.

En este apartado se van a describir los dispositivos utilizados durante la investigación y su situación en la red.

6.1.1. Dispositivos utilizados

Para realizar el estudio, se utilizaron un servidor, un ordenador de desarrollo, una raspberry, una placa NodeMCU y un router.

El servidor se utilizó para mantener los lectores de tráfico conectados, obtenerlos datos sobre los paquetes entrantes y cargar estos datos en la base de datos. En este estudio se utilizó como servidor un ordenador portátil Toshiba Satellite C650. Aunque este dispositivo no es un servidor de gran potencia, se utilizó ya que esta computadora estaba disponible por lo que no requería una gran inversión financiera y poseía la potencia necesaria para realizar las pruebas requeridas por el estudio. En este dispositivo se instalaron Ubuntu 18.04 y un servidor Tomcat. El servidor se mantuvo en actividad constante durante una semana completa, entre el final del desarrollo y el inicio de las pruebas del estudio.

El ordenador de desarrollo utilizado fue un portátil HP-Omen 15-ce0XX.

También se utilizó una Raspberry pi 3 (Raspberry, 2019). Este dispositivo es un miniordenador con poca potencia y sin memoria interna para el cual se necesita disponer de una tarjeta microSD en la que esté instalado el sistema operativo a utilizar. La Raspberry pi ha sido considerada un dispositivo IoT por otros proyectos e. g. (García Muelas, 2018). Dado que no dispone de la potencia de otros ordenadores y su moderado precio, que permite su utilización en este estudio, también se ha considerado la Raspberri PI 3 como un dispositivo IoT y se ha utilizado como tal en las pruebas.

La placa de desarrollo utilizada es la nodeMCU. Es “una placa de desarrollo basada en el ESP8266” (Del Valle Hernández, 2018), utilizada en proyectos en los que se necesita una

conexión a internet sin ser un requisito primordial el de disponer de mucha potencia, por ejemplo, tener una estación meteorológica en casa (Andreu_Rius, 2017).

Por último, se ha utilizado un router Mitrastar HGW-2501GN-R2 para gestionar la red.

6.1.2. Disposición en la red

En la red, nos encontramos, router, con la IP 192.168.1.1. En segundo lugar, tenemos el ordenador de desarrollo en la IP 192.168.1.38 y en la 192.168.1.43 en wifi y ethernet respectivamente. Por último, el servidor dispone de la IP 192.168.1.37.

Los dispositivos que realizarán los ataques de las pruebas, nodeMCU y Raspberry, tienen las IPs 192.168.1.34 y 192.168.1.44 respectivamente.

La figura 13 muestra visualmente la disposición de la red.

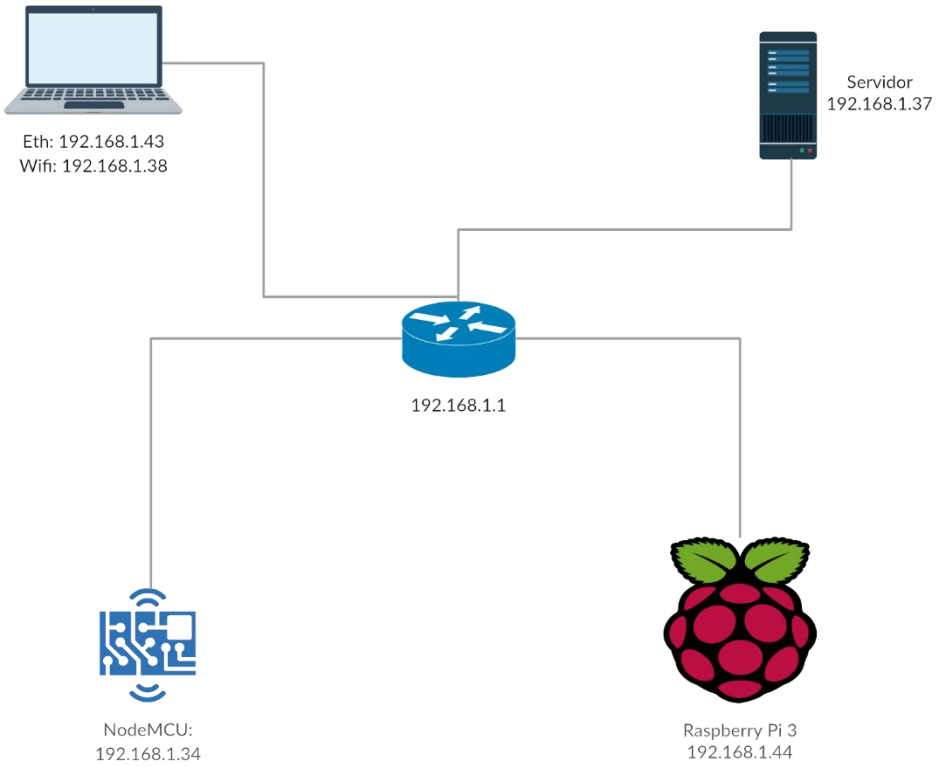

Figura 13 Diagrama de red del laboratorio. Fuente: Elaboración propia

6.2. Pruebas

Este apartado expone el diseño de las pruebas realizadas para la obtención de datos y conclusiones sobre la capacidad de los dispositivos IoT para realizar un ataque DDoS. Las pruebas están diseñadas para comprobar la capacidad de estos dispositivos de realizar

un tipo de ataque concreto. Se examinan siete tipos de ataque, en los que se varía el tipo de ataque en su protocolo, objetivo y el dispositivo atacante. La tabla 12 detalla los dispositivos y tipos de ataque examinados en cada prueba. Los datos y conclusiones obtenidos tras la realización de las pruebas se presentan en las secciones 7 y 11.

Tabla 12

Pruebas propuestas para el estudio.

ID-Prueba	Dispositivo atacante	Tipo de ataque examinado
PE-01	Raspberry Pi 3	Ataque UDP – Puerto determinado
PE-02	Raspberry Pi 3	Ataque TCP – SYN Flood, IP Fija
PE-03	Raspberry Pi 3	Ataque TCP – SYN Flood, IP Dinámica
PE-04	Raspberry Pi 3	ICMP Flood – Ping de la muerte
PE-05	Raspberry Pi 3	Ataque UDP – Puerto variable
PE-06	NodeMCU	Ataque UDP – Puerto determinado
PE-07	NodeMCU	Ataque TCP - IP fija

Nota. Fuente: Elaboración propia

6.2.1. PE-01: Ataque UDP a un puerto determinado desde la Raspberry PI 3

El objetivo de esta prueba es comprobar la capacidad de la Raspberry Pi 3 para realizar un ataque UDP contra el servidor a un puerto específico. Para ello se ha decidido utilizar un código Python de GitHub, inicialmente desarrollado para realizar ataques UDP a puertos variables y modificado para realizar ataques UDP a un puerto fijo (Ha3Mrx, 2018). Se ha dispuesto que el ataque tenga las siguientes características:

- Puerto 700 UDP
- Duración del ataque: 5 Minutos

6.2.2. PE-02: Ataque TCP – SYN Flood con IP fija desde Raspberry Pi 3

El objetivo de la segunda prueba es comprobar la capacidad del dispositivo Raspberry Pi 3 para realizar una multitud de peticiones SYN-TCP al servidor de pruebas. Para ello, se ha decidido utilizar el código de GitHub de fffaraz (Fallahi, 2014), diseñado en un

principio para enviar peticiones SYN-TCP con una IP variable y modificado para utilizar una IP fija en su lugar. Se ha establecido que la prueba tenga las siguientes características:

- Puerto 80 TCP
- Duración del ataque: 5 Minutos

6.2.3. PE-03: Ataque TCP – SYN Flood, IP dinámica desde Raspberry Pi 3

Con esta prueba se pretende determinar la capacidad del dispositivo Raspberry Pi 3 para enviar peticiones SYN-TCP al servidor de pruebas utilizando una IP dinámica. Utilizar una IP dinámica permite mandar paquetes con IPs variables e inalcanzables, lo que puede provocar la saturación del servidor con menos potencia que la necesaria para lograr la saturación del servidor utilizando una IP fija. Para realizar esta prueba se ha decidido utilizar el código de GitHub usado en la prueba anterior (PE-02), pero sin ninguna modificación. Las características definidas para la realización de la prueba son:

- Puerto 80 TCP
- Duración del ataque: 6 Minutos

6.2.4. PE-04: Ping de la muerte desde Raspberry Pi 3

El objetivo de esta prueba es comprobar la capacidad del dispositivo Raspberry Pi 3 de realizar un Ping de la muerte. Este tipo de ataque realiza peticiones ICMP a modo de ping. Debido a la alta cantidad de paquetes enviados puede llegar a saturar la tarjeta de red del servidor. El código utilizado en esta prueba es un código Python de GitHub (it-forensics, 2014). Para la realización de esta prueba se han establecido las siguientes características.

- Duración del ataque: 5 Minutos

6.2.5. PE-05: Ataque UDP a un puerto variable Raspberry Pi 3

El objetivo de esta prueba es comprobar la capacidad del dispositivo Raspberry Pi 3 para realizar un ataque UDP a un puerto variable. Este ataque es similar al realizado en la prueba PE-01, con la diferencia de que, en lugar de enviar todos los paquetes al mismo puerto, manda cada paquete a un puerto diferente. Esto provoca que el ataque sea más difícil de detectar por los sistemas utilizados para ello. Para la realización de la prueba se

ha utilizado el mismo código GitHub utilizado en la prueba PE-01 pero sin realizar ninguna modificación y se han establecido las siguientes características (Ha3MrX,2018):

- Duración del ataque: 5 Minutos

6.2.6. PE-06: Ataque UDP a un puerto determinado desde NodeMCU

Con esta prueba se quiere comprobar la capacidad del dispositivo NodeMCU de realizar un ataque UDP al servidor enviando paquetes a un puerto fijo. El funcionamiento de este ataque es igual que el de la primera prueba (PE-01). Sin embargo, en este caso se utiliza una placa de desarrollo, la NodeMCU, para realizar el ataque, en lugar de un miniordenador. El código utilizado en esta prueba se ha desarrollado en C a partir de la documentación del ESP8266 (Grokhotkov, 2017a). Las características definidas para las pruebas de este ataque son:

- Puerto 8554 UDP
- Duración del ataque: 5 Minutos

6.2.7. PE-07: Ataque TCP – SYN Flood con IP fija utilizando desde NodeMCU

El objetivo de esta prueba es comprobar la capacidad del dispositivo NodeMCU de realizar un ataque TCP a un servidor utilizando una IP fija. El funcionamiento de este ataque es igual al del ataque comprobado en la prueba PE-02, con la diferencia de que en esta prueba el ataque lo realiza una placa de desarrollo en lugar de la Raspberry Pi 3. Para la realización de esta prueba se ha desarrollado un código en C a partir de la documentación “Client”, que se encuentra en la misma página web que la documentación del ESP8266 utilizada para la prueba anterior (Grokhotkov, 2017b). Las características definidas para la realización de la prueba son las siguientes:

- Puerto 80 TCP
- Duración del ataque: 5 Minutos

6.3. Aplicación Qlik

Para el análisis de los resultados obtenidos en las pruebas, se ha utilizado la aplicación Qlik. Qlik proporciona al usuario la posibilidad de usar una multitud de gráficas para visualizar los datos. Dado que Qlik necesita conectarse a elementos del sistema, la aplicación Qlik también se instaló en el laboratorio. Qlik se encuentra instalada en el ordenador de desarrollo y cuenta con acceso a la base de datos del servidor.

Las gráficas descritas en este apartado serán utilizadas para el análisis de resultados (sección 7). Aunque diferentes combinaciones son posibles, en este estudio se ha limitado el número de gráficas utilizadas a nueve que se han considerado más útiles para la obtención de conclusiones. Las gráficas seleccionadas permiten al usuario medir la cantidad de paquetes y bits recibidos por unidad de tiempo e identificar la IP de proveniencia de estos paquetes, a qué puerto están dirigidos y los protocolos utilizados en ellos.

Para medir la **cantidad de paquetes y de bits** recibidos por unidad de tiempo, se han generado gráficas lineales. Las figuras 14 y 15 muestran la suma de bits recibidos por unidad de tiempo y la suma de paquetes recibidos por unidad de tiempo.

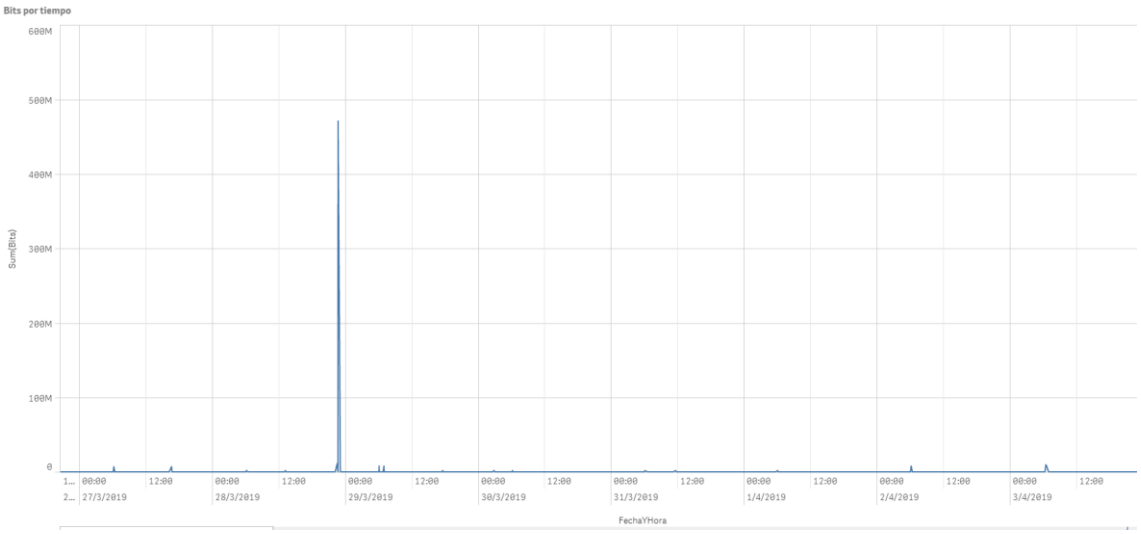

Figura 14 Bits por tiempo. Fuente: Elaboración propia

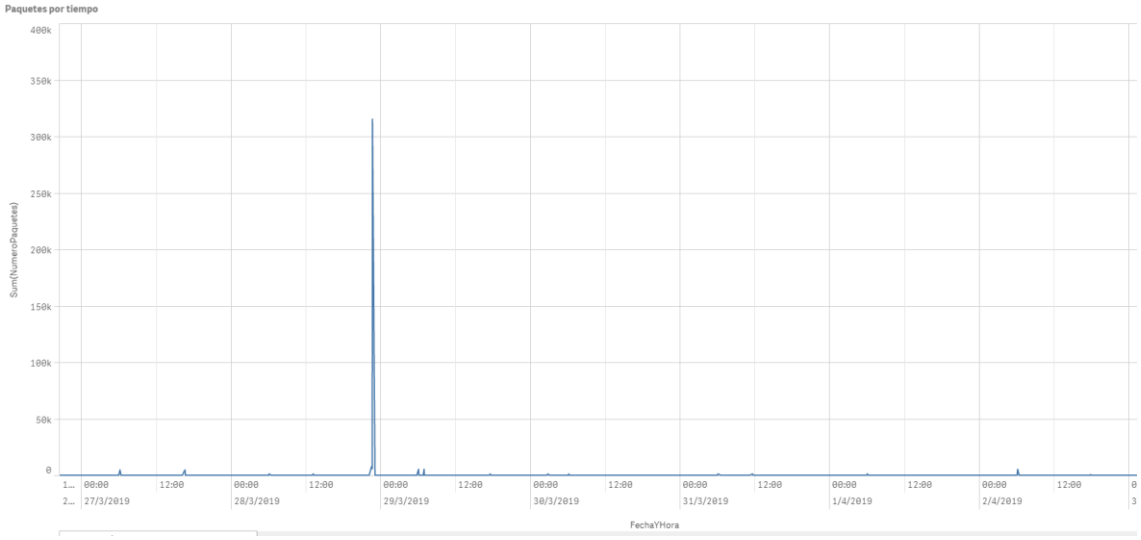

Figura 15 Paquetes por tiempo. Fuente: Elaboración propia

Conocer la IP de procedencia de los paquetes y de los bits permite al usuario identificar un flujo anómalo de datos desde una IP. Para su representación gráfica se han utilizado gráficas de barras. Las figuras 16 y 17 muestran en su eje horizontal las IPs de origen y en el vertical los bits y los paquetes respectivamente.

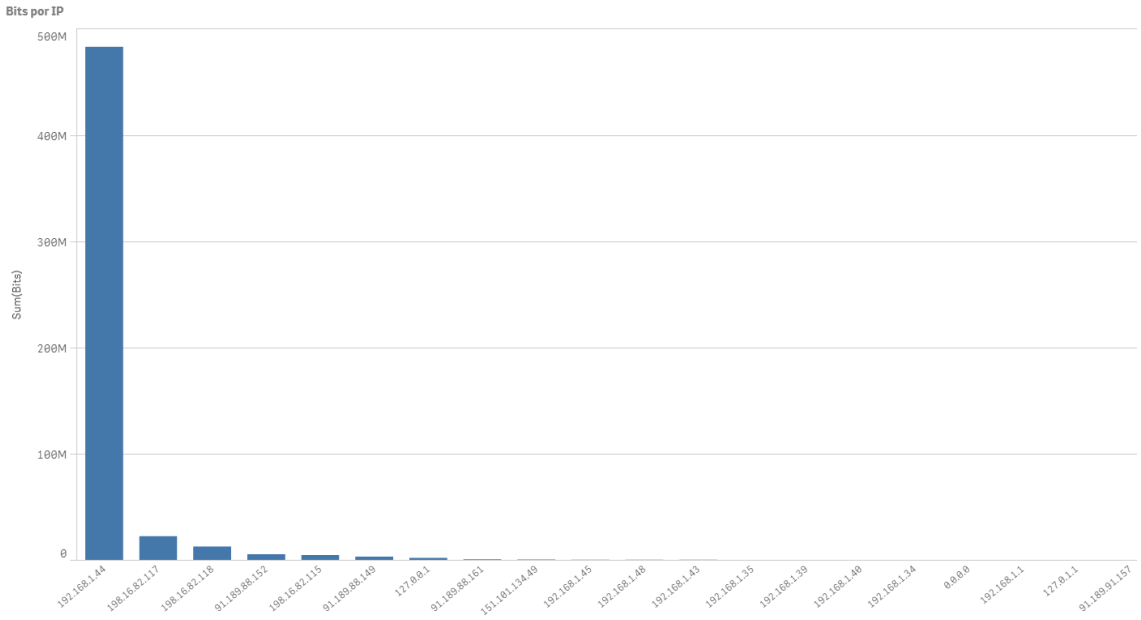

Figura 16 Bits por IP. Fuente: Elaboración propia

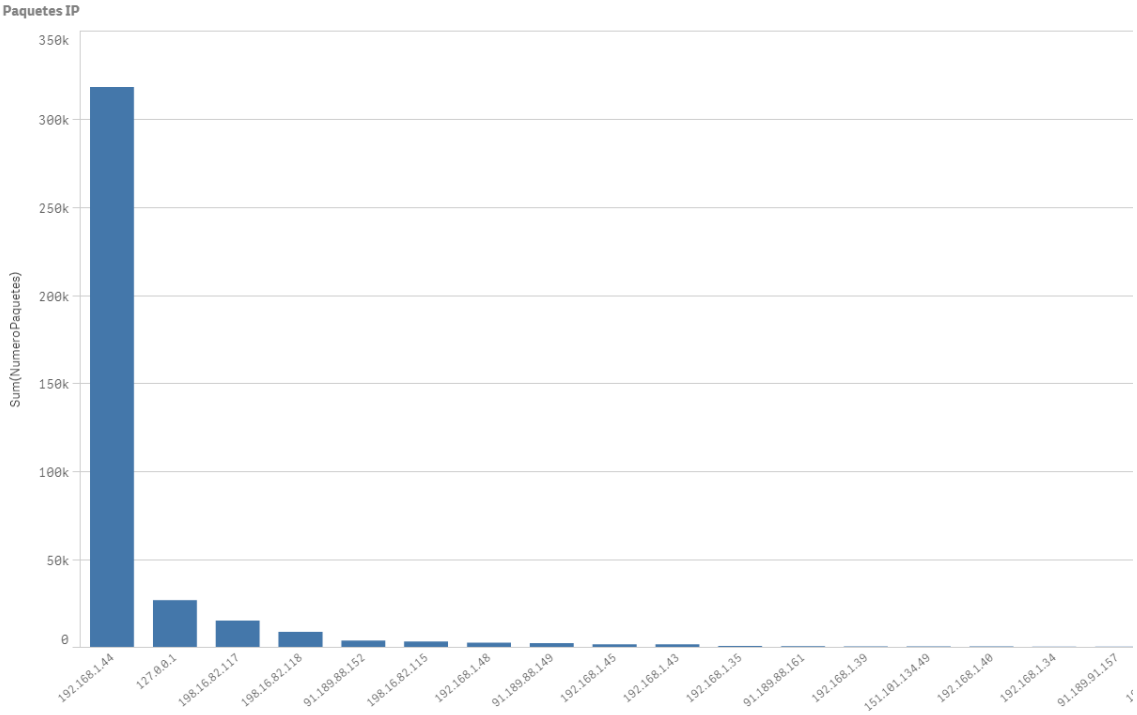

Figura 17 Paquetes por IP. Fuente: Elaboración propia

Conocer el puerto de destino de los paquetes recibidos es útil para el usuario ya que permite identificar hacia qué puerto se dirige un ataque DDoS para identificar una estrategia con el fin detenerlo. Para su representación, se ha decidido generar gráficas de tarta que muestran el porcentaje de bits y paquetes entrantes en cada puerto, como se puede observar en las figuras 18 y 19

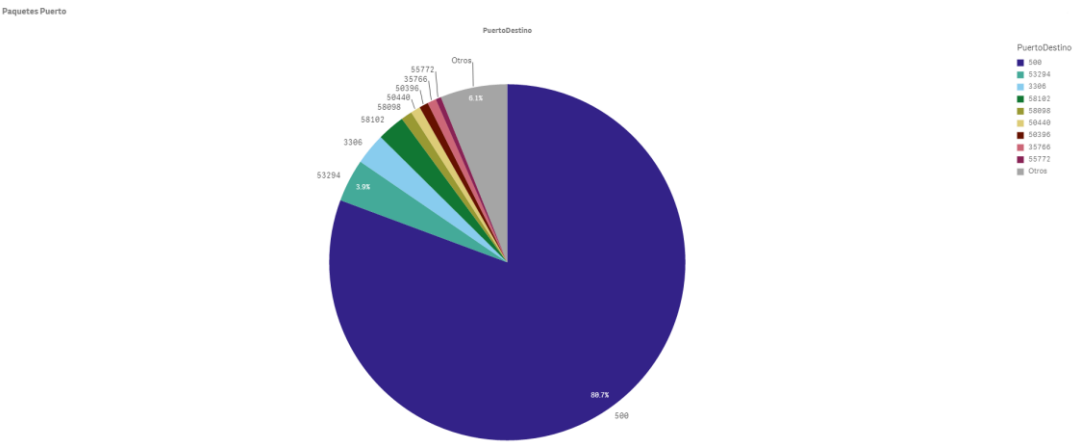

Figura 18 Paquetes por puerto. Fuente: Elaboración propia

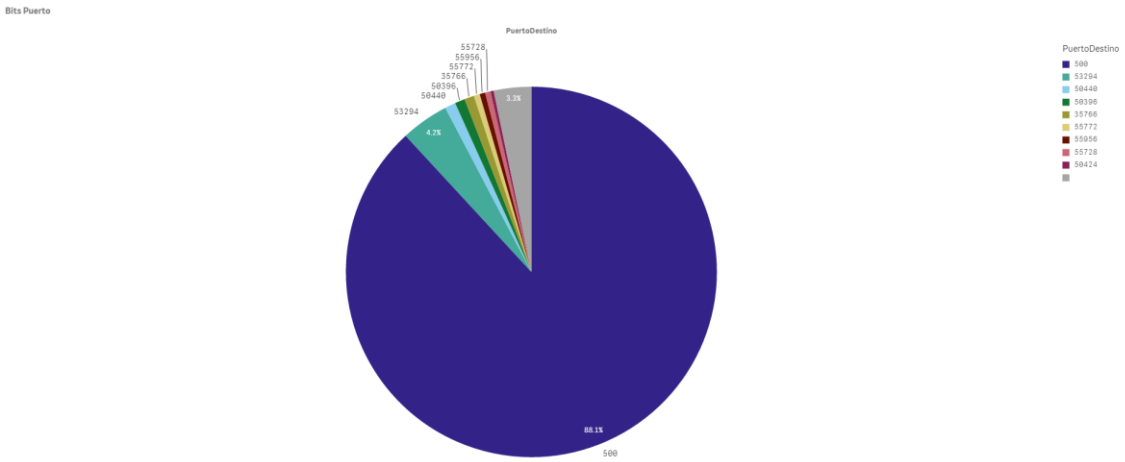

Figura 19 Bits por puerto. Fuente: Elaboración propia

En cuarto lugar, para identificar la forma del ataque se han considerado importantes los protocolos de los paquetes y bits recibidos y se han generado dos gráficos de barras con los tres protocolos considerados en el experimento – ICMP, TCP y UDP - y el número total de bits y de paquetes recibido por protocolo, como se muestra en las figuras 20 y 21. Para complementar estos dos gráficos, se ha añadido un gráfico de bloques mostrando el número paquetes por puerto y por protocolo (figura 22). Esta representación permite al usuario comparar el número de paquetes que llegan a los distintos puertos teniendo en cuenta el protocolo.

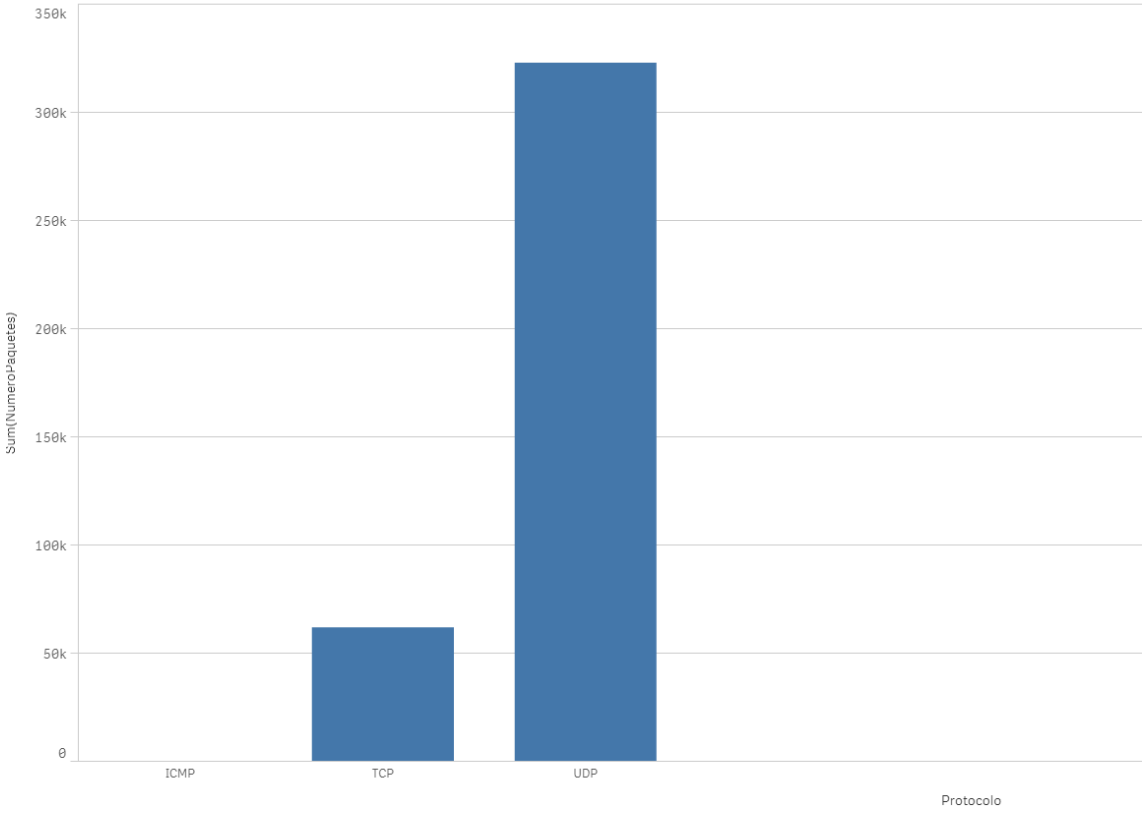

Figura 20 Paquetes por protocolo. Fuente: Elaboración propia

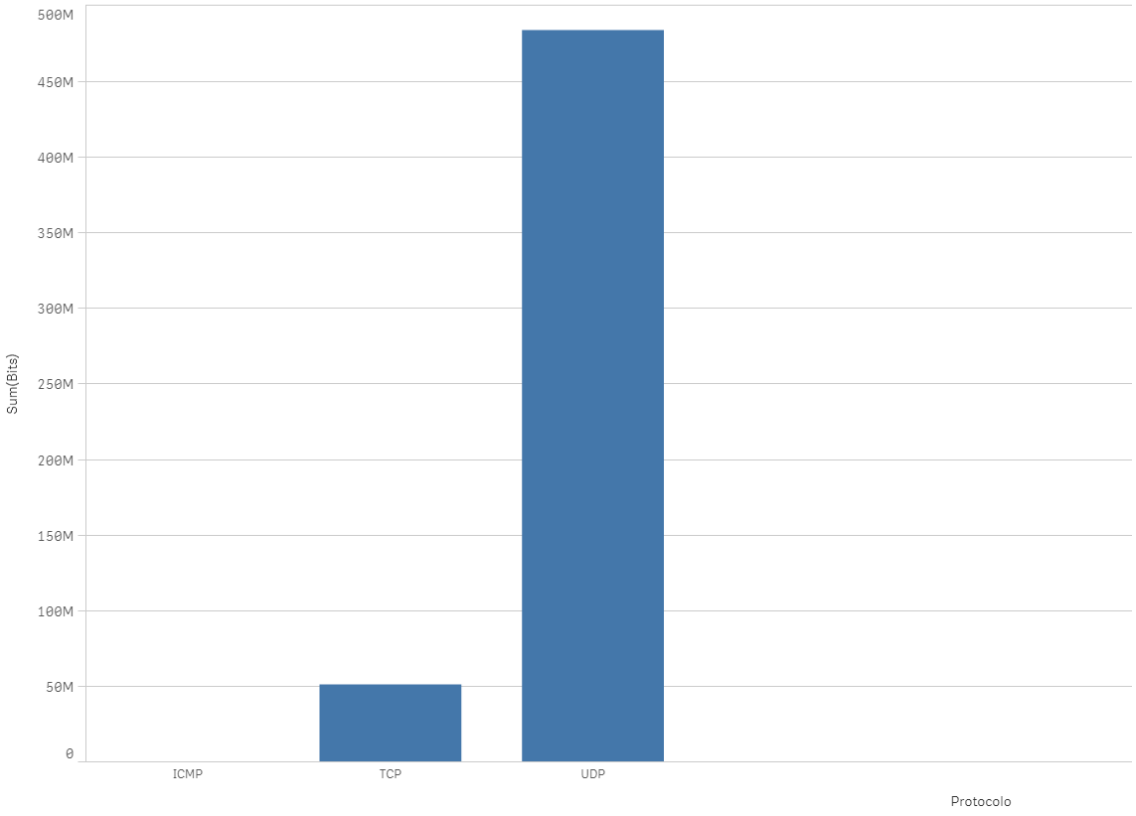

Figura 21 Bits por protocolo. Fuente: Elaboración propia

Figura 22 Paquetes por protocolo y puerto. Fuente: Elaboración propia

Finalmente, también se han generado tablas dinámicas que permiten al usuario combinar la información en el sistema para realizar análisis más específicos o detallados. Las tablas dinámicas incluyen información sobre las IPs de origen, el número de paquetes recibidos y su puerto de destino, y la suma de bits recibidos en cada puerto.

7. RESULTADOS DE LA INVESTIGACIÓN ATAQUES DDoS CON IoT

Esta sección introduce el análisis de los resultados de las pruebas presentadas en el apartado 6.2. Para realizar el análisis se han utilizado los datos obtenidos de Qlik y Wireshark. Wireshark se utiliza para realizar comprobaciones en caso de que las conclusiones obtenidas con Qlik no fueran lo esperado o hubiera dudas, realizando un segundo ataque contra el ordenador de desarrollo. Las conclusiones de este análisis se presentan en la sección 11.

7.1. PE-01 Ataque UDP a un mismo puerto

El objetivo de esta prueba era comprobar la capacidad del dispositivo Raspberry Pi 3 de realizar un ataque UDP a un puerto determinado. Para analizar los resultados de la prueba se midió la potencia del ataque en bits por minuto, que se muestra en la gráfica lineal (Figura 23). La imagen muestra la cantidad de bits recibidos por el servidor durante las veinticuatro horas anteriores a la prueba y durante la prueba. Durante la prueba, que se realizó a las 20 horas, se pudo apreciar un aumento notorio de bits por minutos recibidos, hasta 500 millones de bits por minuto.

Los picos de bits que se aprecian a las 19:55 se deben a que antes de realizar la prueba con los parámetros establecidos, se lanzó una prueba previa para comprobar el estado de la red.

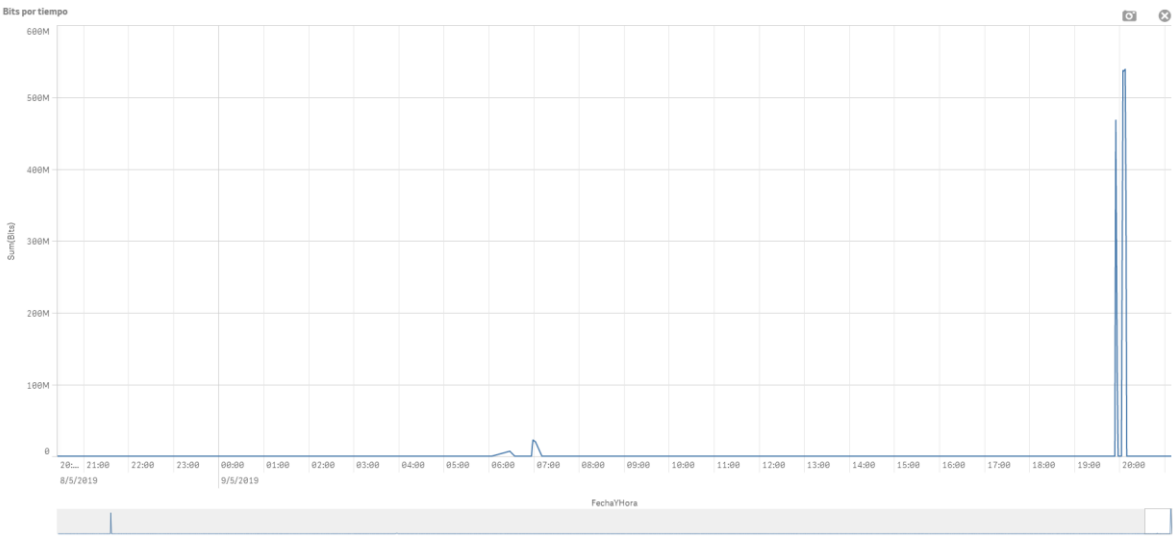

Figura 23 Bits por tiempo PE-01. Fuente: Elaboración propia

Sin embargo, esta gráfica no proporciona la información suficiente como para comprobar el desarrollo y los resultados de la prueba. Conocer el número de bits por minuto que recibe el servidor no permite demostrar que el programa funcione ni que los bits recibidos hayan sido generados por el ataque programado. Tampoco permite determinar que el ataque haya utilizado paquetes UDP ni identificar el puerto a los que han sido enviados esos paquetes. Para visualizar y analizar estos datos, se han utilizado dos gráficas adicionales, que permiten conocer la procedencia y tipo de paquetes de esos bits y su puerto de destino.

Para examinar que los bits recibidos han sido generados por el ataque y que el código funcione, se ha utilizado una gráfica representando la cantidad de bits recibidos por IP (Figura 24). En esta gráfica se puede observar cómo casi la totalidad de los bits recibidos en las 24 horas representadas en la gráfica anterior pertenecen a la IP 192.168.1.44, que es la IP asignada a la Raspberry Pi para el desarrollo de estas pruebas (ver apartado 6.1.2).

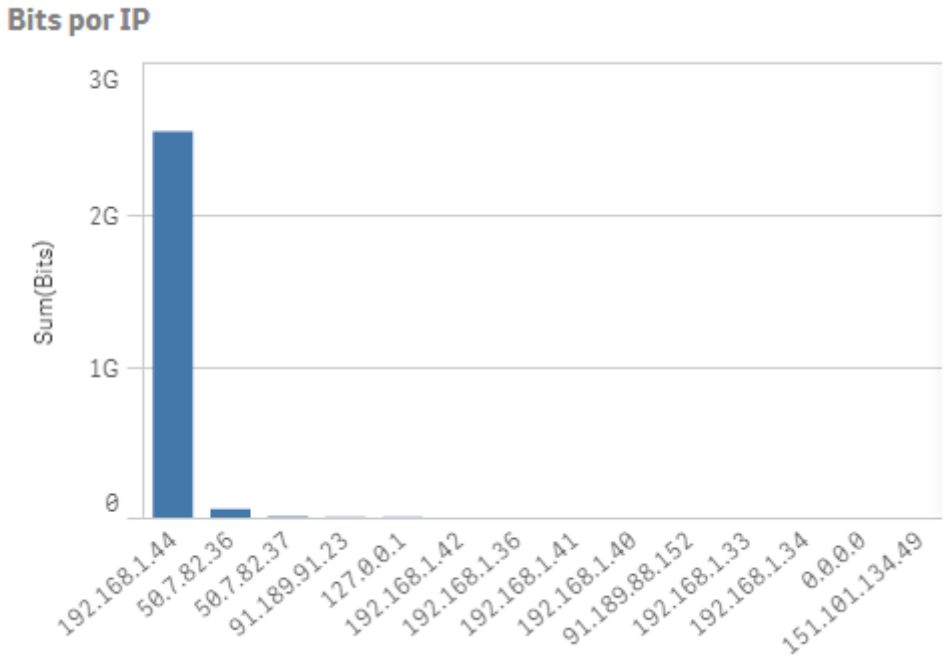

Figura 24 Bits por IP PE-01. Fuente: Elaboración propia

Finalmente, para comprobar que los paquetes son UDP y que son recibidos por el puerto especificado en la prueba, se ha utilizado un gráfico de bloques que representa la cantidad de paquetes que han sido recibidos por cada puerto según el protocolo. En la figura 25 se puede observar como la mayoría de los paquetes llegan al puerto 700.

También se observa una gran cantidad de paquetes recibidos por el puerto 600. Esto se debe a la prueba previamente realizada para comprobar el estado de la red.

Figura 25 Paquetes por protocolo y puerto PE-01. Fuente: Elaboración propia

Con la información presentada en estas gráficas se puede confirmar que estos datos son los correspondientes a la prueba realizada. Se han obtenido los siguientes datos sobre la potencia del ataque:

- Número de paquetes enviados por la raspberry: 1780927
- Número de paquetes que ha sido capaz de capturar el Servidor: 1758894
- Bits recibidos por el Servidor: 2.670.001.092 bits = 2.48 Gbits
- Bits por segundo = 8900003.64 = 0.0083Gbit/s

7.2. Ataque PE-02 TCP-SYN Flood, IP fija

Con esta prueba se pretendía comprobar la capacidad del dispositivo Raspberry Pi 3 para realizar una gran cantidad de peticiones SYN-TCP al servidor de pruebas atacando con una IP fija. Para comprobar el funcionamiento de la prueba y analizar los resultados, se ha utilizado una gráfica lineal que muestra el número de paquetes por unidad de tiempo, y dos gráficas de tarta que muestran el porcentaje de bits y paquetes que llegan a cada puerto y que permiten sacar conclusiones sobre el tamaño de los paquetes recibidos.

La figura 26 muestra la gráfica de paquetes recibidos por unidad de tiempo. En ella se observan tres zonas de picos, correspondientes a tres pruebas diferentes, de las cuales la última es la que ha cumplido los patrones especificados para la prueba PE-02 (Ver apartado 6.2.2). Se puede observar claramente el aumento de los paquetes recibidos por minuto durante la realización de la prueba. Se ha elegido representar los datos recibidos por unidad de tiempo utilizando el número de paquetes recibidos en lugar de los bits recibidos, ya que los ataques TCP mandan paquetes más pequeños que los UDP y se aprovechan del protocolo para que con un menor número de bits afecte a la máquina atacada. Por ello, aunque se podrían obtener las mismas conclusiones utilizando una gráfica de bits por unidad de tiempo, éstas no se verían de una forma tan clara.

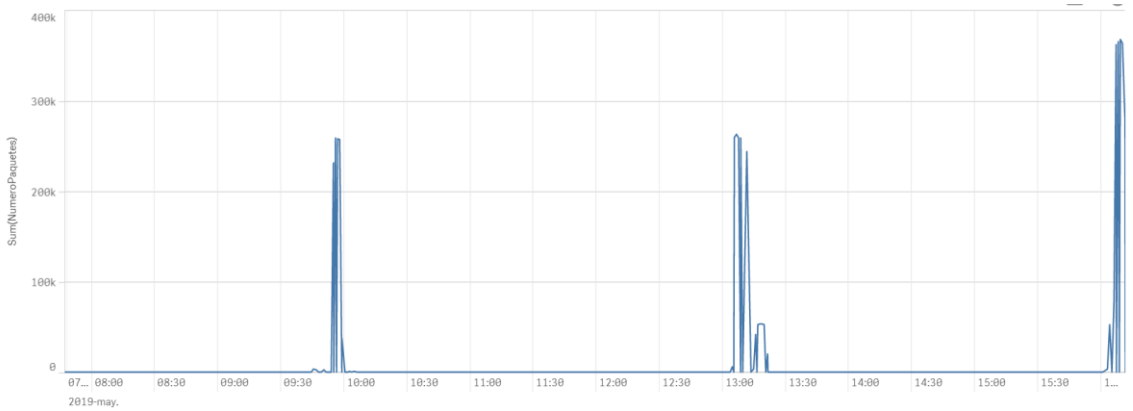

Figura 26 Paquetes por tiempo PE-02. Fuente: Elaboración propia

Para comprobar el tamaño de los paquetes TCP se pueden observar las figuras 27 y 28, que muestran los bits por puerto y los paquetes por puerto respectivamente desde la realización de la primera prueba, PE-01, hasta la finalización de la prueba actual. Se puede observar que solamente un 5,4% de los bits recibidos se han dirigido al puerto 80, el especificado para la segunda prueba, mientras que un 58,3% de los paquetes llegan a ese puerto, indicando el reducido tamaño de los paquetes. Estas figuras pueden ser comparadas con las de la primera prueba, correspondiente al puerto 700 que recibe 63,9% de los bits, pero solo el 24,4% de los paquetes.

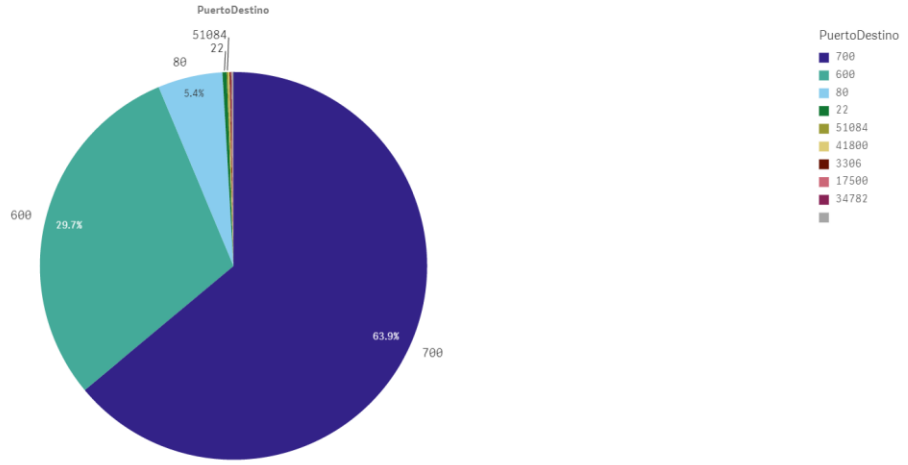

Figura 27 Bits por puerto PE-02. Fuente: Elaboración propia

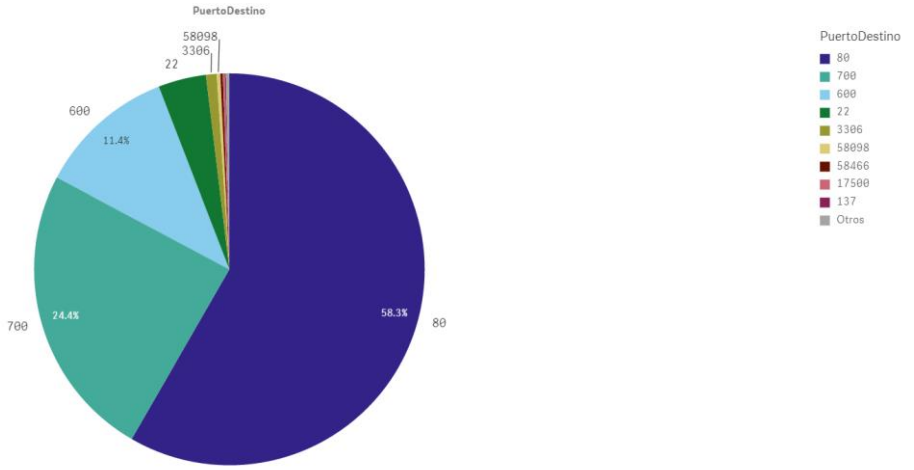

Figura 28 Paquetes por puerto PE-02. Fuente: Elaboración propia

Los valores numéricos obtenidos en este ataque son los siguientes:

- Número de paquetes enviados por la raspberry: 455093
- Número de paquetes que ha sido capaz de capturar el Servidor: 1536821
- Bits recibidos por el Servidor: 82.507.136 bits = 0.0768 Gbits
- Bits por segundo = 275023.79 = 0.000256Gbit/s

Al analizar los datos de este ataque se detectó que el servidor había capturado más paquetes que los enviados por la Raspberry en el ataque. Esto se debe al procedimiento "The three way handshake", que provoca que el servidor al conseguir responder a un paquete sea respondido y lea los dos paquetes en vez de uno, y a que el servidor no es capaz de soportar la cantidad de paquetes recibidos y retransmite algunos al acabar su

tiempo de espera. La retransmisión de los paquetes se puede ver en la figura 29, que es una captura de Wireshark hecha al investigar este comportamiento.

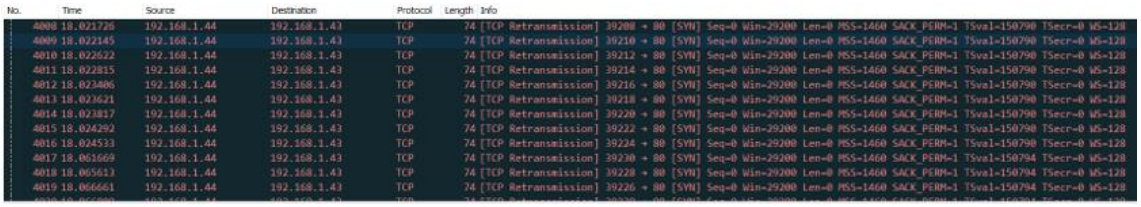

Figura 29 Captura Wireshark TCP Retransmision PE-02. Fuente: Elaboración propia.

7.3. PE-03 Ataque TCP – SYN Flood, IP dinámica

Con esta prueba se pretendía comprobar la capacidad del dispositivo Raspberry Pi 3 para realizar una gran cantidad de peticiones SYN-TCP, pero esta vez atacando con una IP dinámica. Este tipo de ataque, al tener que generar las IPs de forma aleatoria, es menos eficiente a la hora de enviar paquetes, La figura 30 muestra la cantidad de paquetes recibidos por unidad de tiempo durante y después del ataque. Se puede apreciar un claro descenso en el número de paquetes una vez concluido el ataque, y que pese al utilizar una IP dinámica, la Raspberry Pi 3 es capaz de mandar unos 700 paquetes por minuto.

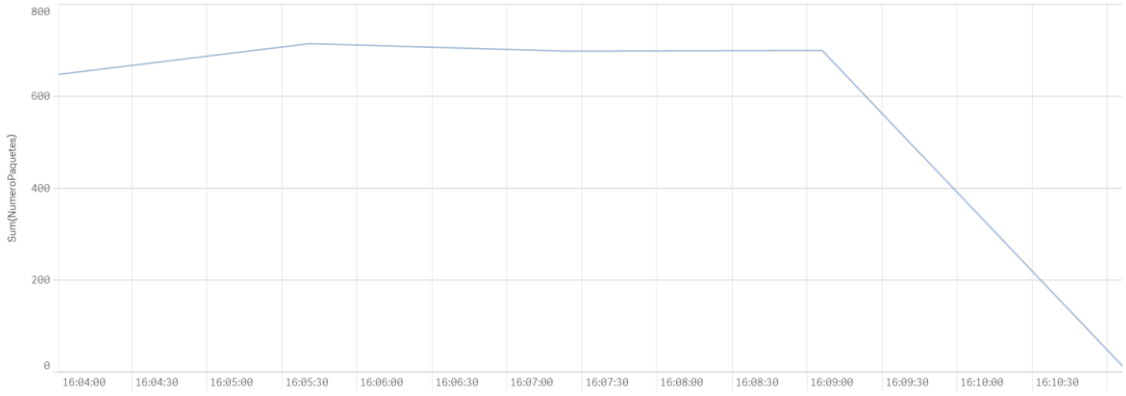

Figura 30 Paquetes por tiempo PE-03. Fuente: Elaboración propia

Si comparamos los 700 paquetes por minuto enviados en este ataque con los 360000 paquetes por minuto enviados cuando se utilizaba una IP fija, se podría decir que el ataque utilizando una IP dinámica va a ser menos eficaz que un ataque TCP desde una IP fija. Sin embargo, utilizar una IP aleatoria permite aprovechar el protocolo TCP para saturar el ordenador más fácilmente que con una IP fija. Al utilizar IPs inalcanzables, al hacer el "the three way handshake", no hay ninguna posibilidad de que la respuesta del servidor llegue a un ordenador, y de que este conteste con el ACK. Por ello, el servidor queda a la

espera del ACK y continúa enviando peticiones para recibirlo. Esto posibilita bloquear el servidor enviando una menor cantidad de paquetes.

Una vez comprobada la capacidad de enviar paquetes al servidor, se examinaron las IPs desde las que se realizó el ataque y cuántos paquetes y bits fueron enviados por cada IP. Utilizando el mismo filtro de tiempo que el usado en la gráfica anterior (figura 30), se estudiaron las IPs de origen, el número de paquetes y de bits y el puerto de destino. La figura 31 muestra un extracto de la tabla de Qlik con estos parámetros. Las IPs utilizadas por este ataque son extrañas y poco comunes.

Para verificar que esas eran las IPs generadas por el ataque y que no habían sido generadas por un fallo del servidor al leerlas, se generó un nuevo ataque; esta vez contra el ordenador de desarrollo, para analizarlas con Wireshark. Las IPs de este segundo ataque eran parecidas a las del ataque anterior, y se concluyó que el hecho de que las IPs fueran tan poco comunes se debía a las características del ataque y no a un error de lectura por parte del servidor. Un extracto de este segundo ataque se puede observar en la figura 32.

IpOrigen	Sum(NumeroPaquetes)	PuertoDestino	Sum(Bits)
Totales	**2773**		**110920**
1.20.46.200	1	80	40
1.37.118.2	1	80	40
1.72.215.16	1	80	40
1.105.247.13	1	80	40
1.165.140.170	1	80	40
1.167.168.0	1	80	40
1.236.195.131	1	80	40
1.243.8.129	1	80	40
1.253.2.111	1	80	40
1.254.71.85	1	80	40
2.113.77.144	1	80	40
2.126.231.237	1	80	40
2.134.95.208	1	80	40
2.157.78.112	1	80	40
2.184.140.239	1	80	40

Figura 31 Tabla dinámica PE-03. Fuente: Elaboración propia

No.	Time	Source	Destination	Protocol	Length	Info
6155	344.076902	175.68.192.245	192.168.1.43	TCP	60	7134 → 80 [SYN] Seq=0 Win=2640 Len=0
6156	344.077950	21.12.126.76	192.168.1.43	TCP	60	6814 → 80 [SYN] Seq=0 Win=8592 Len=0
6157	344.108450	41.41.36.252	192.168.1.43	TCP	60	5082 → 80 [SYN] Seq=0 Win=4896 Len=0
6158	344.189334	175.35.1.244	192.168.1.43	TCP	60	7755 → 80 [SYN] Seq=0 Win=1367 Len=0
6159	344.332245	98.163.38.44	192.168.1.43	TCP	60	8564 → 80 [SYN] Seq=0 Win=3307 Len=0
6160	344.348017	92.25.169.53	192.168.1.43	TCP	60	4003 → 80 [SYN] Seq=0 Win=5469 Len=0
6161	344.475843	255.53.229.236	192.168.1.43	TCP	60	3434 → 80 [SYN] Seq=0 Win=8567 Len=0
6162	344.508342	61.175.184.48	192.168.1.43	TCP	60	2112 → 80 [SYN] Seq=0 Win=2968 Len=0
6163	344.653578	66.32.231.69	192.168.1.43	TCP	60	3939 → 80 [SYN] Seq=0 Win=8682 Len=0
6164	344.784027	51.255.0.167	192.168.1.43	TCP	60	2572 → 80 [SYN] Seq=0 Win=2250 Len=0
6165	344.784464	123.178.40.239	192.168.1.43	TCP	60	7032 → 80 [SYN] Seq=0 Win=2488 Len=0

Figura 32 Captura Wireshark IPs Generadas PE-03. Fuente: Elaboración propia

Los datos numéricos obtenidos tras la realización de esta prueba fueron los siguientes:

- Número de paquetes enviados por la raspberry: 5000
- Número de paquetes que ha sido capaz de capturar el Servidor: 2773
- Bits recibidos por el Servidor: 118.920 bits = 0.0001 Gbit/s

7.4. PE-04 Ataque ICMP Flood – Ping de la muerte

En esta prueba se utilizó la Raspberry Pi 3 para realizar un Ping de la muerte, un tipo de ataque que envía peticiones ICMP para saturar la tarjeta de red del servidor. Durante la realización de la prueba se produjo un resultado inesperado: el servidor no era capaz de leer los paquetes enviados. Se consideró que se podía estar cortando la conexión de la red del servidor, y se procedió a generar un nuevo ataque contra el ordenador de desarrollo y utilizar Wireshark para analizar lo que sucedía. El ordenador tardó menos de tres segundos en cortar la tarjeta de red. En la figura 33 se puede ver la alerta que devolvía Wireshark al ejecutar el ataque.

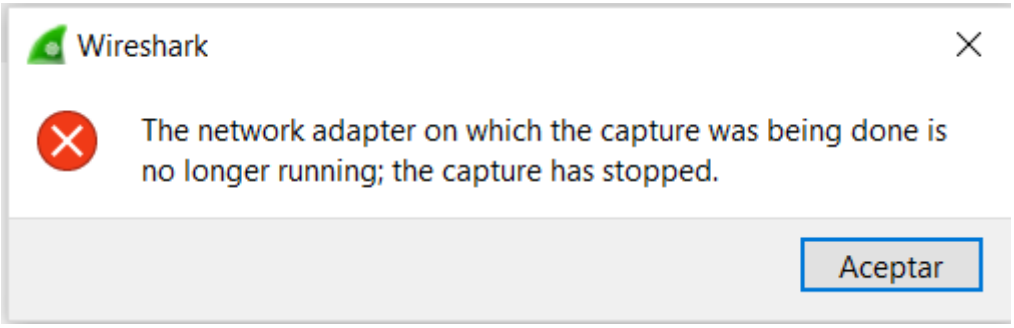

Figura 33 Notificación de caída de red de Wireshark PE-04. Fuente: Elaboración propia

7.5. PE-05 Ataque UDP puerto variable

En esta prueba se realizó un ataque UDP a un puerto variable. Esta prueba es muy parecida a la prueba PE-01, con la excepción de los puertos a los que se dirige el ataque. De hecho, ambas pruebas utilizan el mismo código base con ligeras modificaciones. Por ello, en el análisis de resultados se decidió comparar algunos aspectos de ambas pruebas. Al igual que en la prueba PE-01, se utilizó la gráfica de bits por tiempo para medir la potencia del ataque y la gráfica de bloques para examinar la cantidad de paquetes recibidos por cada puerto.

La gráfica de bits por tiempo indica que el ataque llega a enviar una cantidad ligeramente superior a quince millones de bits por minuto (Figura 34). Esta cantidad corresponde solamente al 3% de los bits generados por el ataque UDP con puerto fijo. Sin embargo, estos datos no significan que este ataque sea menos eficaz, ya que es un ataque mucho más difícil de identificar por los anti-DDoS y sigue mandando una gran cantidad de bits por minuto.

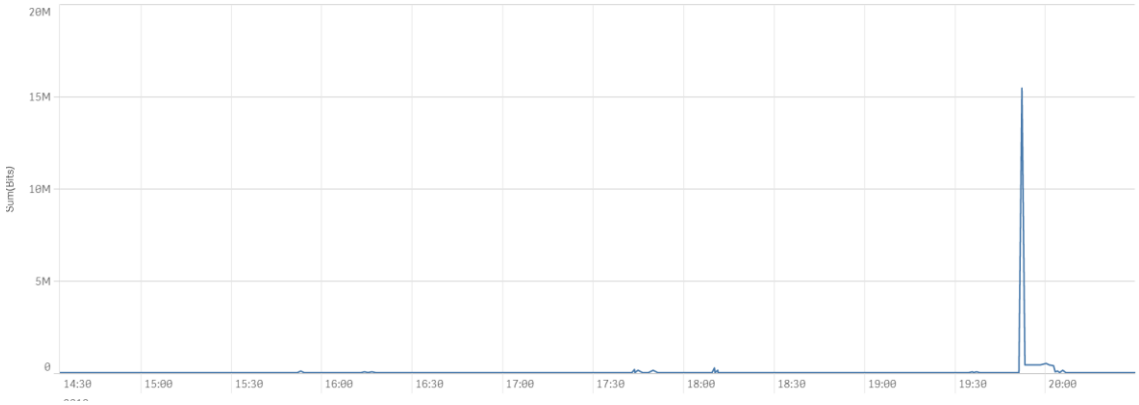

Figura 34 Bits por tiempo PE-05. Fuente: Elaboración propia

Ambas pruebas no se diferencian solamente en la cantidad de bits enviados. La mayor diferencia es, como queda indicado anteriormente, que en esta prueba se ataca a un puerto variable en lugar de a uno fijo. Esto se puede ver claramente en el gráfico de bloques. Mientras que en la prueba PE-01 el gráfico aparece muy plano con dos puertos dominantes, en el gráfico de esta prueba se observan multitud de puertos que han recibido un solo paquete (Ver figura 35).

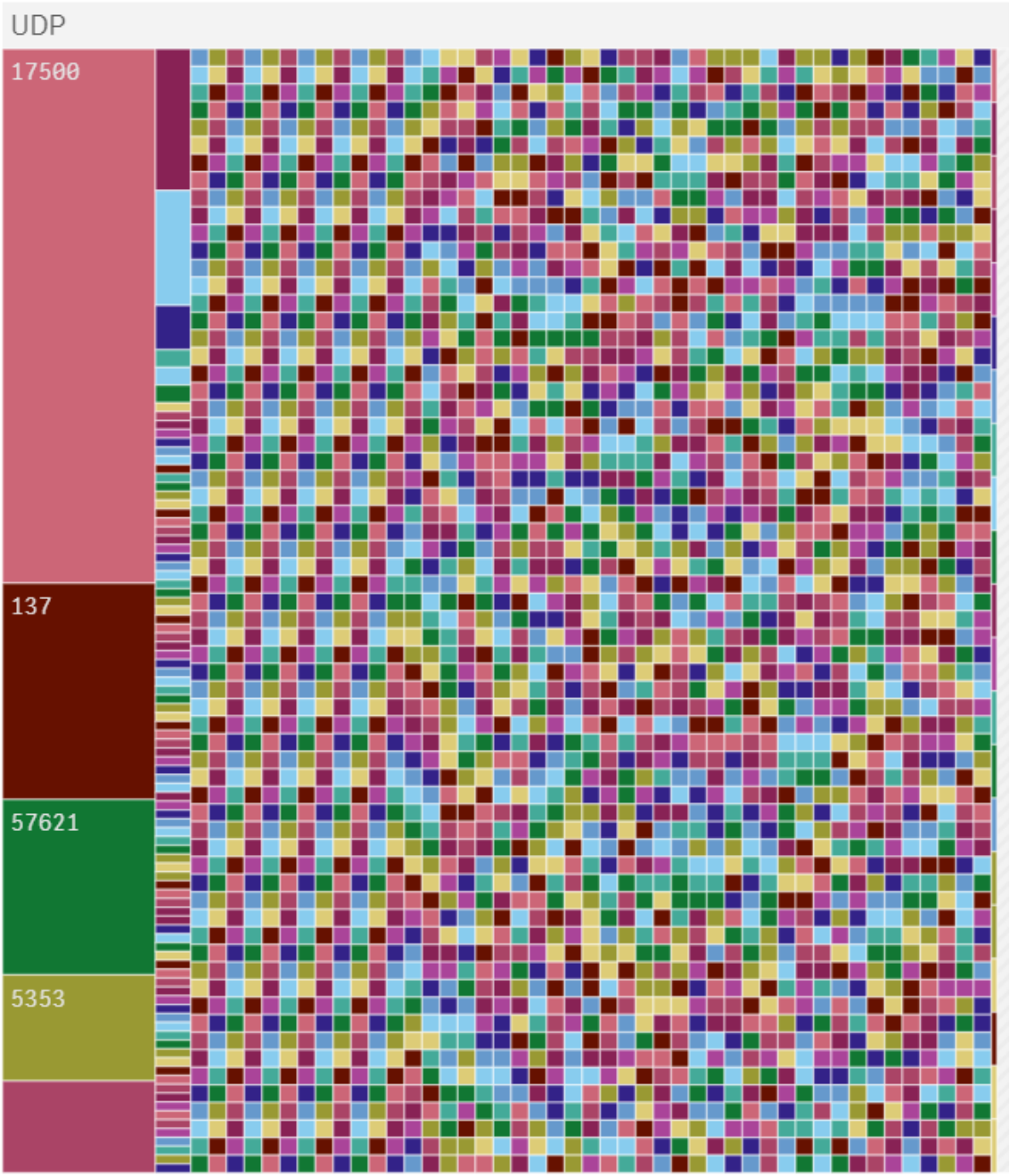

Figura 35 Paquetes por protocolo y puerto PE-05. Fuente: Elaboración propia

La ventaja que tiene este tipo de ataque es que si se realizara desde una botnet en lugar de desde un solo dispositivo sería muy difícil de detectar, ya que una botnet nos permitiría enviar los paquetes desde diferentes IPs y que los paquetes llegan a diferentes puertos.

Los datos obtenidos tras los análisis de este ataque son los siguientes:

- Número de paquetes enviados por la Raspberry: 1391909
- Número de paquetes que ha sido capaz de capturar el Servidor: 10230

- Bits recibidos por el Servidor: 15.525.150 bits = 0.0145 Gbits
- Bits por segundo = 51750.5= 0,00004819Gbit/s

La gran pérdida de paquetes detectada se debe a que algunos de los puertos del servidor estaban cerrados o no admitían paquetes UDP y los paquetes no llegaron a su destino.

7.6. PE-06 Ataque UDP a un puerto determinado

Esta prueba, al igual que la prueba PE-01, realiza un ataque UDP a un puerto determinado, pero utilizando una placa Node MCU en lugar de la Raspberry Pi 3. Esta prueba es la primera de las realizadas con la placa Node MCU. Se espera que estos ataques sean menos potentes que los generados con la Raspberry, debido a la diferencia de potencia, tamaño y precio de ambos dispositivos. Al igual que en la prueba PE-01, se utilizó una gráfica de paquetes por tiempo para comprobar la potencia del ataque y una gráfica de bits por IP para comprobar que el tráfico viniese de la placa.

Para que el ataque fuera evidente en la gráfica de paquetes recibidos por unidad de tiempo, se filtró el tiempo mostrado en la gráfica para que solo mostrara esta prueba, y no las realizadas anteriormente. En esta gráfica (ver figura 36) se puede observar que la placa consiguió que el servidor recibiera más de cuarenta mil paquetes por minuto en algunas instancias del ataque.

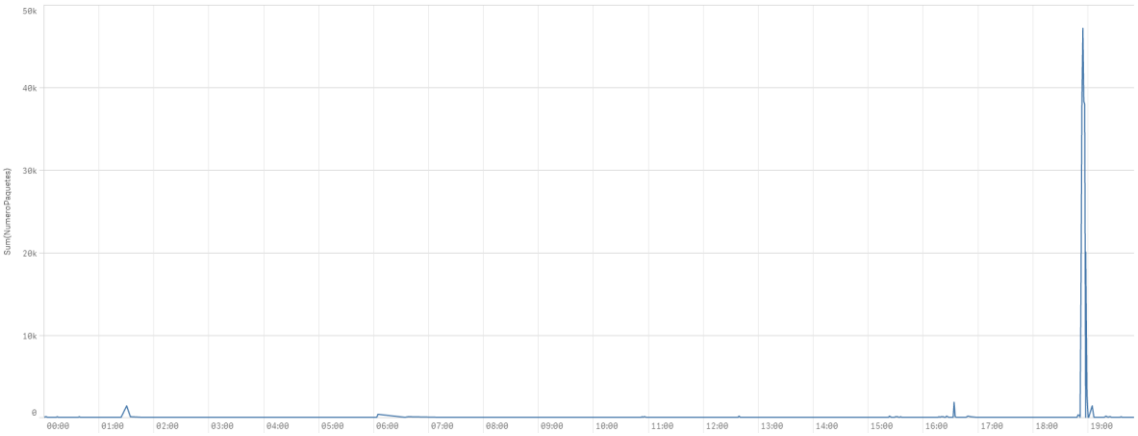

Figura 36 Paquetes por tiempo PE-06. Fuente: Elaboración propia

Para comprobar que el tráfico fuera causado por la placa se comprobó la gráfica de bits por IP limitado a ese tiempo. En este caso, en la figura 37, se ve como la IP 192.168.1.34, que es la asignada a la placa NodeMCU en el apartado 6.1.2, genera la mayor cantidad de tráfico aportando alrededor de once millones de bits.

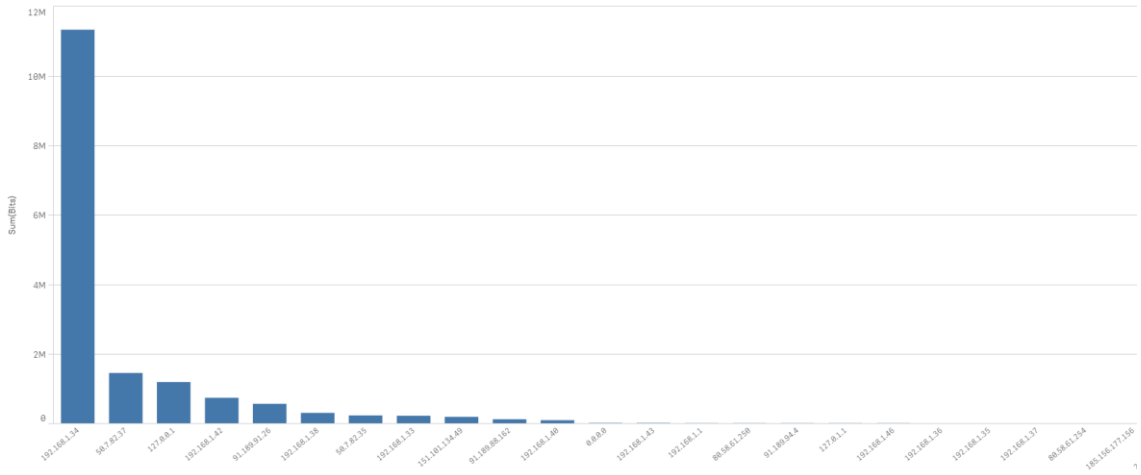

Figura 37 Bits por IP PE-06. Fuente: Elaboración propia

Los datos numéricos obtenidos con esta prueba son los siguientes:

- Número de paquetes enviados por la NodeMCU: 440127
- Número de paquetes que ha sido capaz de capturar el Servidor: 197729
- Bits recibidos por el Servidor: 11270553 bits = 0.0105 Gbits
- Bits por segundo = 37568.51= 0.00003499 Gbit/s

7.7. PE-07 Ataque TCP con IP fija

En esta prueba se realizó un ataque TCP al servidor desde la placa NodeMCU utilizando una IP fija. Durante la realización de la prueba se detectaron algunas interrupciones en el envío de paquetes debidos a la capacidad máxima que tiene la placa para mantener conexiones TCP. Sin embargo, el ataque consiguió cierta ralentización del servidor a la hora de cargar la web que mantiene el servidor.

Para evaluar los paquetes enviados de esta prueba se ha decidido usar como referencia la prueba anterior, usando la gráfica de división de paquetes por puerto para compararlas. En la figura 38 se comprueba que el puerto atacado en esta prueba, el 80, recibió un 28.4% de los paquetes en la franja de tiempo analizada, desde el inicio de la gráfica de la imagen (figura 36) y final de la prueba PE-07, mientras que en el puerto asignado a la PE-06, el 8554, el 60.5% de los paquetes.

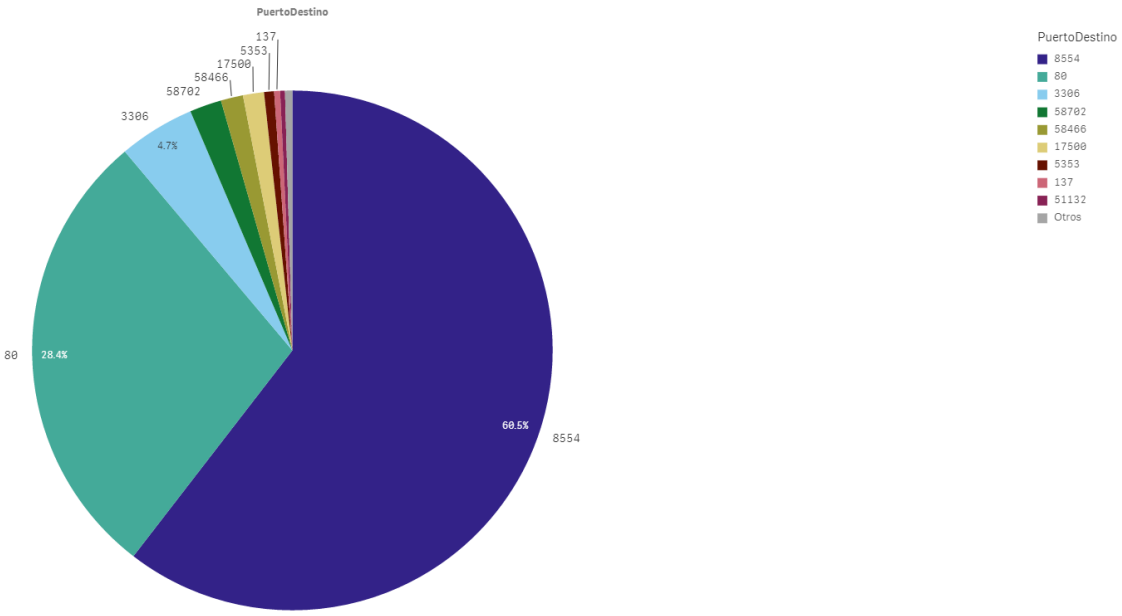

Figura 38 Paquetes por puerto PE-07. Fuente: Elaboración propia

Los valores numéricos obtenidos fueron los siguientes:

- Número de paquetes enviados por la NodeMCU: 23010
- Número de paquetes que ha sido capaz de capturar el Servidor: 91842
- Bits recibidos por el Servidor: 21853 bits = 0.0000203522Gbit/s

Igual que en la PE-02 se puede ver que el servidor detecta más paquetes que los que manda el código de la placa. La justificación a este comportamiento es la misma que en el caso anterior. El "the three way handshake" provoca que el servidor reciba el doble de paquetes y, además, al saturarse, retransmite algunos al terminar el tiempo de espera de estos.

7.8. Resumen de las pruebas

La tabla 13 muestra un resumen de las pruebas, si se pudo obtener los datos de potencia, si es posible realizar ataques DDoS de ese estilo con ese dispositivo, número de dispositivos necesarios valorando 1,5Gbit/s como un pequeño ataque DDoS y otros comentarios sobre la prueba.

Tabla 13

Resumen de las Pruebas.

ID-Prueba	Dispositivo	Tipo de ataque	Potencia	Dispositivos necesarios para 1,5Gbit/s
PE-01	Raspberry Pi 3	UDP- Puerto determinado	0.0083Gbit/s	181
PE-02	Raspberry Pi 3	TCP- IP fija	0.000256Gbit/s	5860
PE-03	Raspberry Pi 3	TCP- IP dinámica	0.0001 Gbit/s	15000
PE-04	Raspberry Pi 3	Ping de la muerte	No se pudieron obtener datos debido a la potencia del ataque	
PE-05	Raspberry Pi 3	UDP- Puerto variable	0,00004819Gbit/s	31.127
PE-06	Node MCU	UDP- Puerto determinado	0.00003499Gbit/s	42.870
PE-07	Node MCU	TCP- IP fija	0.0000203522Gbit/s	73.703

Nota. Fuente: Elaboración propia

Como muestra la tabla anterior todos los ataques han sido detectados por el software desarrollados salvo la prueba PE-04. También como se verá en la sección 11, valorando el número de dispositivos IoT existentes actualmente todos estos tipos de ataques son factibles y con muy poco porcentaje del total de dispositivos.

8. DATOS

En esta sección se presentan los informes utilizados como fuente de datos. Una vez analizados estos datos las conclusiones de este análisis se exponen en la sección 10.

En la tabla 14 se muestran dichos informes, su año de publicación, el método de obtención de datos utilizado, el tamaño y características de su muestra y los datos reportados que se consideraron relevantes para este estudio.

Entre los datos más importantes existentes en los diferentes informes se encuentran el número de ataques, potencia de los ataques y los costes medios de los ataques.

Tabla 14

Informes utilizados.

Autor	Año	Informe	Obtención de Datos	Tamaño de la Muestra	Datos de la Muestra	Datos Importantes
Ponemon Institute	2012	cyber security on the offense	Encuesta	705	Profesionales IT y Ciberseguridad 61% reportan directamente al CIO y 21% al CISO Media de 11 años de experiencia	Media de ataques por sector año 2011: Financiero: 3 ataques Público: 4,1 Sanitario: 2,4 Duración media de ataques: 54 minutos Coste medio 22.000$/minuto
Kaspersky	2014	Global IT Security Risks Survey	Encuesta	3900	27 países Representantes de empresas 54% medianas, grandes y muy grandes empresas	Coste medio de sufrir un ataque DDoS: Grandes empresas 440.000$ Pequeñas empresas 52.000$ Daño reputacional: 38% de los encuestados creen que les afecta
Kaspersky	2015	GLOBAL IT SECURITY RISKS SURVEY 2015	Encuesta	5564	38 países Profesionales IT	Sectores más atacados: Bancos, Telecomunicaciones, Servicios financieros 50% de los encuestados sufrieron ataques
NETSCOUT & Arbor Networks	2018	NETSCOUT Arbor's 13th Annual Worldwide Infrastructure Security Report	Encuesta	390	55% Proveedores de servicios 45% Empresas, gobierno y educación 66,6% Profesionales de ciberseguridad o redes	87% de los proveedores de servicio detectaron ataques DDoS 99% del resto de empresas encuestados reportaron al menos 1 ataque 57% de las empresas afectadas consideran que dañó su reputación
NETSCOUT	2018	NETSCOUT 14th Annual Worldwide Infrastructure Security Report	Encuesta	No especificado		44% de las empresas utilizan anti-DDoS Coste medio de ataque 22.1836,80$
Neustar	2019	The Changing Face of Cyber Attacks (Neustar Defense)	Encuesta	300	Profesionales de ciberseguridad	DDoS Tercera mayor amenaza

Autor	Año	Informe	Obtención de Datos	Tamaño de la Muestra	Datos de la Muestra	Datos Importantes
Neustar	2019	CyberTrheats & trends report (Neustar)	Encuesta	>170	Profesionales de ciberseguridad	DDoS La mayor amenaza
Khalimonenko, Kupreev, & Ilganaev	2017	Los ataques DDoS en el tercer trimestre de 2017	Datos de red capturados por sus sistemas	No especificado	Sistemas de Kaspersky a nivel mundial	
Khalimonenko, Kupreev, & Ilganaev	2018	Los ataques DDoS en el cuarto trimestre de 2017	Datos de red capturados por sus sistemas	No especificado	Sistemas de Kaspersky a nivel mundial	
Khalimonenko, Kupreev, y Badovskaya	2018	Los ataques DDoS en el primer trimestre de 2018	Datos de red capturados por sus sistemas	No especificado	Sistemas de Kaspersky a nivel mundial	
Ibragimov, Kupreev, Badovskaya y Gutnikov	2018	Los ataques DDoS en el segundo trimestre de 2018	Datos de red capturados por sus sistemas	No especificado	Sistemas de Kaspersky a nivel mundial	
Kupreev, Badovskaya y Gutnikov	2018	Los ataques DDoS en el tercer trimestre de 2018	Datos de red capturados por sus sistemas	No especificado	Sistemas de Kaspersky a nivel mundial	

Nota. Fuente: Elaboración propia

9. METODOLOGÍA

El principal objetivo de esta investigación es poder determinar si a las empresas les resulta más rentable contratar un servicio anti-DDoS o ahorrarse el coste, pero corriendo un mayor riesgo. Con este objetivo en mente, la investigación comenzó con la búsqueda de informes sobre el coste que generan estos ataques en Google, buscando "costes de los ataques DDoS", "DDoS attacks cost generated" y otras variaciones con el mismo enfoque.

Las búsquedas mencionadas en el párrafo anterior llevaron a páginas de noticias que mencionan estudios como el IT Security Risks Survey de Kaspersky lab. Esto hizo que se enfocara la búsqueda en este tipo de informes (Intereconomía, 2018).

En el rastreo de estos informes se encontraron otros con la misma temática y forma de distintas empresas e instituciones como "Netscout", "Arbor Networks", "Ponemon Institute" y "Neustar" e. g. (Neustar, 2019a).

Además de analizar la información y datos de los informes encontrados se creyó oportuno tratar de hacer una regresión para intentar predecir la evolución del número de ataques DDoS y se empezaron a buscar bases de datos con ataques DDoS reales. El resultado de esta búsqueda de bases de datos fue el hallazgo de un informe que hacía algo parecido a lo que se buscaba junto con un enlace a su base de datos, pero debido a no residir en unos países determinados la solicitud del acceso a esas bases de datos fue denegada (Moore et al., 2001).

Se encontró otra base de datos a lo largo de este proceso a la que se pudo acceder, pero su uso fue descartado debido a la inconsistencia de su contenido. También se optó por contactar con el Instituto Nacional de Ciberseguridad (INCIBE) para intentar obtener algún tipo de base de datos sobre ataques DDoS a nivel nacional, pero no disponían de la información necesaria.

Por último, tras obtener los datos e información de los informes se decidió realizar un análisis cualitativo comparando todo lo obtenido y evaluándolo.

10. RESULTADOS ANÁLISIS DEL PERJUICIO DE UN ATAQUE DDoS

En esta sección se muestran los resultados divididos en número de ataques, potencia de los ataques, duración de los ataques, el coste provocado y otros datos.

10.1. Número de ataques

Para poder valorar la magnitud y el desarrollo de estos ataques es útil considerar la evolución del número de ataques reportados anualmente. Para ello, se ha decidido comparar los informes de Netscout y Arbor Networks (2018) y de Netscout (2019), que informan sobre el número de ataques realizados.

El informe de Netscout junto con Arbor Networks (2018) en el que se realizó una encuesta a 390 representantes de empresas, de las cuales el 55% eran proveedores de servicios de telecomunicaciones, indica que solo un 8% de estos proveedores reportan haber sufrido menos de un ataque al mes mientras que un 17% reportan detectar más de 500 ataques al mes.

Por otro lado, el siguiente informe realizado por Netscout (2019), del cual se desconocen los datos de los encuestados, muestra un aumento en ambos porcentajes mencionados en el párrafo anterior, siendo un 9% superior en los que reportan menos de un ataque al mes y un 5% superior los que reportan más de 500 ataques al mes.

Para poder analizar correctamente la evolución identificada entre estos informes se utilizó la tabla 15 que representa los datos de ambos informes y se estableció un valor constante como multiplicador para calcular un valor ponderado para evaluar la evolución de los ataques. Este valor es arbitrario y se ha utilizado solamente para valorar el crecimiento o la disminución del número de ataques entre ambos años. Esta evaluación demuestra que existe un crecimiento anual estimado de un 23% en el número de ataques a proveedores de servicios de red, pero no proporciona información sobre el crecimiento exacto de dicha cifra.

Tabla 15

Evolución del número de ataques

	Frecuencia de ataques al mes							
Año	menos de 1	de 1 a 10	de 11 a 20	de 21 a 50	de 51 a 100	de 101 a 500	más de 500	
2017	8%	33%	14%	13%	4%	11%	17%	
2018	17%	23%	7%	17%	3%	12%	22%	
	Valores establecidos							
	0,5	5	15	35	75	200	600	Totales
2017	0,04	1,65	2,1	4,55	3	22	102	135,34
2018	0,085	1,15	1,05	5,95	2,25	24	132	166,485

Nota. Fuente: Elaboración propia, a partir de los datos: Netscout y Arbor Networks (2018); Netscout, (2019)

Sin embargo, no solo es importante conocer la evolución del número de ataques DDoS, sino también a cuántas empresas afectan. Según Netscout y Arbor Networks (2018) el 99% de las empresas encuestadas que no eran proveedoras de servicios de telecomunicaciones, reportan haber recibido al menos un ataque en 2017.

Por otro lado, el informe de la encuesta realizada por Kaspersky (2015) a 5564 profesionales IT de 38 países, indica que un 50% de los negocios ha sufrido algún trastorno debido a ataques DDoS y un 47% de los encuestados afirman que su web fue inhabilitada debido a un ataque. Este mismo informe también menciona que los sectores más atacados ese año fueron los bancos, las telecomunicaciones y los servicios financieros.

Con los datos existentes se puede concluir que una gran proporción de empresas sufren ataques DDoS al año. Además, dado que el número de ataques DDoS aumenta cada año, es posible que el número de empresas afectadas también aumente, lo que incrementaría la amenaza que ya suponen estos ataques.

10.2. Potencia de los ataques

La potencia de los ataques es otra característica que es necesario considerar para evaluar el peligro que suponen estos, ya que un ataque más potente es capaz de causar más daño. Dado que con los datos disponibles no es posible medir la potencia de todos los ataques

al año, se ha decidido hacer un estudio de la evolución de la potencia máxima que ha alcanzado un ataque en cada año.

Una gráfica presentada en el informe de Netscout (2019, p. 44) presenta la mayor potencia de los ataques reportado desde 2008 hasta 2018. Estos valores se han utilizado para examinar la evolución de la potencia de los ataques salvo por el presentado en el 2018, que ha sido modificado. Esta modificación se debe a la existencia de dos importantes ataques sucedidos a principios del 2018, uno a la plataforma GitHub con una potencia de 1,35Tbps el 28 de febrero, y otro con una potencia de 1,7Tbps del cual no se ha podido localizar más información aparte de la potencia (González G., 2018; Neustar, 2019b). Dada la falta de información sobre el segundo ataque, se decidió hacer dos regresiones diferentes, la primera valorando la potencia del ataque a GitHub (1,35Tbps) como potencia máxima del 2018 y la segunda valorando la potencia del ataque de 1,7Tbps como tal.

La tabla 16 muestra los datos utilizados en las regresiones, indicando el año y la potencia más alta reportada en ese año. El aumento de la potencia de los ataques año a año es evidente, lo que también indica la correlación presente entre el año y la potencia. Esta correlación es positiva, de 0.8492 en el caso de valorar como el ataque más potente de 2018 el ataque a GitHub y de 0,8036 en caso de valorar como ataque más potente el otro ataque. Aun sin ser esta correlación de gran fiabilidad debido a la escasez de datos, su presencia indica un posible crecimiento en la potencia de los ataques DDoS, lo que puede crear un mayor peligro para las empresas afectadas por ellos.

Tabla 16

Correlación año y tamaño de ataques.

	Regresión 1:	Regresión 2:
Año	TamañoAtaqueGbps	TamañoAtaqueGbps
2008	40	40
2013	309	309
2014	400	400
2015	500	500
2016	800	800
2017	600	600
2018	1350	1700
Correlación	0,8492	0,8036

Nota. Fuente: Elaboración propia a partir de los datos: Netscout, (2019)

10.3. Duración de los ataques

Ya que los ataques DDoS saturan el servidor durante un determinado periodo de tiempo, su duración es un aspecto clave que hay que considerar para estimar el daño que pueden causar.

Ponemon Institute (2012) marca una media de duración de los ataques DDoS en 2011 de 54 minutos. Por otro lado, los informes trimestrales de la empresa Kaspersky proporcionan la información de los ataques que ellos han detectado gracias a sus sistemas, clasificando los ataques por duración en varios rangos de horas, desde menos de cuatro hasta más de 100 horas y exponiendo también la duración del ataque más largo. En la tabla 17 se muestra la información proporcionada por los informes trimestrales de Kaspersky (Khalimonenko, Kupreev & Ilganaev, 2017; Khalimonenko, Kupreev & Ilganaev, 2018; Khalimonenko, Kupreev y Badovskaya, 2018; Ibragimov, Kupreev, Badovskaya y Gutnikov, 2018; Kupreev, Badovskayay Gutnikov, 2018).

Tabla 17

Porcentajes tamaño de ataques Informes Kaspersky.

año + trimestre	ataque más largo	<4	"5-9"	"10-19"	"20-49"	"50-99"	100-140	>100
2017 Q3	215horas	76,09%	10,33%	9,50%	3,73%	0,30%	0,40%	0,02%
2017 Q4	146horas	76,76%	8,28%	10,20%	4,65%	0,08%	0,02%	0,01%
2018 Q1	297 horas	80,73%	10,73%	4,93%	2,82%	0,52%	0,11%	0,14%
2018 Q2	258 horas	69,49%	14,01%	10,05%	5,25%	0,96%	0,11%	0,12%
2018 Q3	239 horas	86,94%	5,49%	3,79%	3,07%	0,50%	0,09%	0,10%

Nota. Fuente: Elaboración propia a partir de los datos: Khalimonenko et al., (2017); Khalimonenko et al., (2018); Khalimonenko et al., (2018b); Ibragimov et al., (2018); Kupreev et al., (2018)

Para poder comparar los datos de los informes de Kaspersky (tabla 17) con los proporcionados por Ponemon Institute es necesario utilizar la misma unidad. Ya que los informes de Kaspersky no proporcionan información sobre el tiempo medio de los ataques en cada categoría, se ha decidido estimar estos tiempos. Con el objeto de conseguir esos valores y evitar sobrevalorar o infravalorar la duración de los ataques reportados por la empresa, se ha decidido realizar dos computaciones, la primera valorando los rangos por su valor mínimo y utilizando 0,2 como valor en el rango de

"menos de 4 horas" (tabla 18) y la segunda utilizando valores intermedios en cada rango y 0,5 en el rango de "menos de 4 horas" (tabla 19).

Tabla 18

Evaluación media de duración de los ataques (1).

Valor multiplicador	0,2	5	10	20	50	100	101	Total
2017 Q3	0,15218	0,5165	0,95	0,746	0,15	0,4	0,0202	2,93488
2017 Q4	0,15352	0,414	1,02	0,93	0,04	0,02	0,0101	2,58762
2018 Q1	0,16146	0,5365	0,493	0,564	0,26	0,11	0,1414	2,26636
2018 Q2	0,13898	0,7005	1,005	1,05	0,48	0,11	0,1212	3,60568
2018 Q3	0,17388	0,2745	0,379	0,614	0,25	0,09	0,101	1,88238

Nota. Fuente: Elaboración propia a partir de los datos: Khalimonenko et al., (2017); Khalimonenko et al., (2018); Khalimonenko et al., (2018b); Ibragimov et al., (2018); Kupreev et al., (2018)

Tabla 19

Evaluación media de duración de los ataques (2).

Valor multiplicador	0,5	7	15	35	75	120	160	Total
2017 Q3	0,38045	0,7231	1,425	1,3055	0,225	0,48	0,032	4,57105
2017 Q4	0,3838	0,5796	1,53	1,6275	0,06	0,024	0,016	4,2209
2018 Q1	0,40365	0,7511	0,7395	0,987	0,39	0,132	0,224	3,62725
2018 Q2	0,34745	0,9807	1,5075	1,8375	0,72	0,132	0,192	5,71715
2018 Q3	0,4347	0,3843	0,5685	1,0745	0,375	0,108	0,16	3,105

Nota. Fuente: Elaboración propia a partir de los datos: Khalimonenko et al., (2017); Khalimonenko et al., (2018); Khalimonenko et al., (2018b); Ibragimov et al., (2018); Kupreev et al., (2018)

De estos cálculos se obtiene que la mínima media del tamaño de ataques obtenida en estos trimestres es de 1,88 horas en el tercer trimestre de 2018 (tabla 18), mientras que la media más alta fue en el segundo trimestre de ese mismo año con una media de 5.71 horas (tabla 19). Si se comparan ambos casos con los 54 minutos que de los que hablaba Ponemon Institute (2012) se observa un claro aumento de la duración de los ataques DDoS.

Considerando el gran aumento en la duración de los ataques DDoS desde el año 2012 hasta el 2018, se puede concluir que es altamente probable que la duración de estos ataques continúe aumentando en los próximos años.

10.4. Coste de daños provocados por ataques

Los ataques DDoS causan diferentes tipos de costes para las empresas, incluyendo los costes de maquinaria estropeada, el pago de servicios para recuperarse de un ataque, la pérdida de ingresos durante el tiempo que duran los ataques, la disminución de la reputación y la subida de precio de los seguros. Algunos de estos costes no son fácilmente valorables económicamente y debido al alto grado de confidencialidad de todos los costes relacionados con ataques informáticos los datos de los informes tampoco son fiables completamente. Sin embargo, sí que se puede hacer una estimación de la totalidad de los costes para valorar la gravedad de estos ataques.

Las conclusiones de Ponemon Institute (2012) en su informe fueron que el coste medio de un ataque DDoS era de 22.000$/minuto. Considerando la duración media de estos de 54 minutos, el coste medio de los ataques reportados en este informe es de 1.188.000$. Sin embargo, tras valorar la información de otros informes se ha concluido que esta cifra es desorbitada y no corresponde al coste medio causado por un ataque DDoS.

Kaspersky (2014) en su "Global IT Security Risks Survey 2014", que entrevistó a 3900 representantes de empresas de 27 países de las cuales el 54% eran grandes y medianas empresas, habla de un coste medio de 52000$ para las empresas pequeñas y 440.000$ para las empresas grandes. Este coste medio es más razonable que el presentado por Ponemon Institute (2012) y está también en concordancia con el informe de Netscout (2019), que indica un coste medio por ataque de 221.836,80$.

Además de estos costes económicos que pese a no ser tan desorbitados como los reportados por Ponemon Institute (2012) constituyen un gasto importante, las empresas afectadas por ataques DDoS sufren otros daños que no son fácilmente valorables de forma económica. Para estimar estos costes, se han utilizado informes que utilizan encuestas para obtener la perspectiva de los afectados por este tipo de ataques sobre los daños ocasionados.

De los 3900 encuestados por Kaspersky (2014) en 2014, el 38% de los negocios creía que un ataque DDoS dañaba la reputación de su empresa, un 29% reportó que le afectó a su calificación de crédito y un 26% reportó un aumento en las primas de sus seguros en ese año. En el 2015, la misma empresa realizó un estudio similar, en el cual de los 5564 encuestados, un 27% decía que su mayor miedo de recibir un ataque DDoS era la perdida

de reputación, otro 27% la posible pérdida de ingresos y oportunidades y un 15% la posibilidad de perder clientes actuales como resultado de un ataque (Kaspersky, 2015). Se puede concluir que más de un cuarto de las empresas afectadas por estos ataques sufren otros costes además de los directamente causados por el ataque y que la pérdida de reputación y de clientes, y no solo la pérdida de ingresos, son considerados importantes.

Netscout y Arbor Networks (2018) indican que un 57% de las empresas afectadas por ataques DDoS creían que afectaban a su reputación, un 11% reportaron un aumento en las primas de los seguros y otro 11% se vieron afectadas por una pérdida de empleados. En el informe realizado el año siguiente por Netscout (2019) estos porcentajes variaron a 37%, 37% y 38% respectivamente. Como se ve entre estos informes, hay una disminución de las empresas que creen que disminuye su reputación, pero aumenta considerablemente el número de ellas al que les incrementan las primas de seguros y les afectan las pérdidas de empleados que provoca una gran pérdida de capital humano. La conclusión que se puede obtener de estos informes es, al igual que en el párrafo anterior, que no solo los costes directos tienen que ser considerados importantes.

Por lo tanto, sufrir un ataque DDoS puede suponerle multitud de pérdidas tanto económicas como reputacionales, pudiendo incluso llegar a quebrar una empresa según su tamaño (Kaspersky, 2017). El alto coste de los ataques, junto con su ya mencionado crecimiento en número y potencia, indica la necesidad cada vez mayor de que las empresas tomen medidas y contraten servicios anti-DDoS.

10.5. Otros datos

En este apartado se exponen otros datos obtenidos de los informes que se han considerado relevantes para la conclusión final pero que no se pueden enmarcar en ninguno de los apartados anteriores.

Uno de estos datos es la consideración por parte de las empresas sobre los ataques DDoS. Ponemon Institute (2012) mostraba en su informe que las defensas anti-DDoS eran consideradas las segundas más importantes después de los antivirus y que los ataques DDoS eran considerados una amenaza prioritaria a la hora de mitigarse. El informe de Kaspersky (2014) dice que el departamento IT del 23% de los negocios consideran la prevención de los ataques DDoS como prioridad máxima. En el 2017 el DDoS fue considerado como la mayor amenaza por las empresas que respondieron a la encuesta de

Netscout y Arbor Networks (2018). También Neustar (2019a) posiciona los ataques DDoS como la tercera mayor amenaza y en su posterior informe como la mayor amenaza, como se explicó en el apartado 3.2 de la sección 3 (Neustar, 2019b). Por lo tanto, estos ataques ya son considerados importantes por multitud de empresas.

Por último, es importante saber el estado actual de uso de anti-DDoS por parte de las empresas; en este apartado también hay una gran variación entre unos informes y otros, desde un 71% de uso según Ponemon Institute (2012) hasta 40% según Kaspersky (2014). Estas grandes variaciones es posible que se deban a los tipos de empresas encuestadas por ambos informes. Por ejemplo, los informes de Netscout y Arbor (2018) y Netscout (2019) que indican que un 88% los proveedores de servicios web utilizan este tipo de servicios frente a un 44% del resto de empresas.

Por otro lado, el informe de Kaspersky (2015) indica que tan solo un 56% de los profesionales de IT consideran rentable invertir en servicios anti-DDoS. La conclusión que se puede obtener es que, aunque la mayor parte de las empresas y profesionales IT sí que considera prioritario el defenderse de los ataques DDoS, aún existen multitud de ellas que no lo hacen y que podrían sufrir muchos daños en caso de ser objetivo de un ataque.

Por último, hay que mencionar el coste de los servicios anti-DDoS. Estos costes se adaptan a la empresa contratante, y pueden variar desde ir incluidos en la factura telefónica hasta costar más de 12.000€ al mes. Esto, viendo los costes que llegan a provocar los ataques DDoS, no supone una gran inversión para una empresa.

11. CONCLUSIÓN Y FUTURAS INVESTIGACIONES

11.1. Conclusiones

11.1.1. DDoS con IoT

Con esta investigación se quería comprobar la capacidad de los dispositivos IoT de participar en un ataque DDoS. Las pruebas realizadas en la sección 7 han permitido comprobar que los dispositivos utilizados (la Raspberry pi 3 y la placa NodeMCU) sí que son capaces de realizar este tipo de ataques, y que esta capacidad es muy elevada considerando su tamaño, su coste económico y sus características.

Por ejemplo, un ataque UDP a un mismo puerto con una Raspberrry Pi 3 puede alcanzar los 0,0083Gbit/s. Para realizar un pequeño ataque de 1,5Gbit/s serían necesarias tan solo 181 Raspberries. Teniendo en cuenta que en marzo de 2017 ya se habían vendido más de 12,5 millones de Raspberries y que gran parte de ellas están constantemente funcionando, un hacker que logre controlar un reducido porcentaje del total de Raspberries podría controlar una botnet de gran potencia para atacar a empresas que no tengan contratados servicios anti-DDoS. En el caso menos potente de la Raspberry Pi 3, el ataque UDP con puertos variables, serían necesarios 31127 dispositivos lo que equivale a un 0,25% del total de Raspberries vendidas en 2017 (Martínez, 2017).

Respecto a las NodeMCU, para realizar un ataque de UDP o TCP de 1,5Gbits/s se necesitarían 42.870 y 73.703 dispositivos respectivamente. Si se considera que la potencia obtenida se puede amplificar por diversos métodos no estudiados en este documento y que también se podrían utilizar otros dispositivos IoT, de los cuales en 2017 había 8400 millones, se puede apreciar el peligro que puede conllevar la falta de protección de estos dispositivos frente al control no deseado por parte de los hackers (Neustar, 2019a).

Dado que los dispositivos IoT cuentan con poca protección frente a los hackers y pueden ser fácilmente utilizados para realizar ataques DDoS mediante botnets, sería beneficioso incrementar la seguridad de los dispositivos por parte de sus proveedores para para dificultar la utilización de estos dispositivos en dichos ataques.

11.1.2. Uso de Anti-DDoS

El segundo objetivo de esta investigación era el de examinar si a las empresas les resulta rentable contratar un servicio anti-DDoS. Para valorarlo, es necesario considerar tanto el coste del servicio anti-DDoS como todos los costes, directos e indirectos, causados por

los ataques DDoS, que son numerosos e incluyen costes económicos y reputacionales e incluso pérdida de clientes (ver sección 10). A pesar de que contratar un sistema anti-DDoS no garantiza que la empresa no sea víctima de un ataque DDoS, sí que puede evitar o reducir los efectos del ataque y los daños ocasionados.

Para decidir si contratar un servicio anti DDoS es rentable, también es necesario considerar que el importe de los servicios anti-DDoS se adapta a las necesidades de cada empresa, pudiendo ir desde un precio prácticamente incluido en la factura de la línea telefónica hasta más de 12.000€ al mes. Dado que en los informes evaluados al menos un 50% de los encuestados había sufrido un mínimo de un ataque DDoS al año y que el coste medio de un ataque DDoS supone alrededor de 221.836,80$, se puede estimar un coste medio mensual por sufrir uno de estos ataques de 9.243,2$ (Netscout y Arbor Networks, 2018; Netscout, 2019; Neustar, 2019a).

$$\frac{0{,}5 \text{ x } 221.836{,}80}{12} = 9.243{,}2\$ \qquad (1)$$

Observando este cálculo (1), teniendo en cuenta que los 9.243,2$/mes, equivalente a unos 8.340€, sería el coste asumido por una empresa al no estar protegida y que los costes de contratar servicios anti-DDoS se adaptan a las características de cada una de las empresas, se concluye que invertir en servicios anti-DDoS es una decisión que deberían tomar todas las empresas.

11.1.3. Conclusiones generales

Para resumir, teniendo en cuenta la relación entre el número de dispositivos IoT y el aumento del número de ataques DDoS, el continuo crecimiento de los dispositivos IoT, su vulnerabilidad ante los hackers y su capacidad de realizar ataques DDoS y los altos costes que sufrir un ataque DDoS causa a empresas e instituciones, es recomendable para cualquier tipo de empresa invertir en un servicio anti-DDoS que se ajuste a sus necesidades y que pueda protegerla de la creciente amenaza que suponen estos ataques (INCIBE, 2019; Perakovic et al., 2015).

11.2. Futuras investigaciones

Existen diversas investigaciones que se pueden hacer a partir de este proyecto.

Empezando por la investigación que no se ha podido realizar en este proyecto, inferir

futuros ataques DDoS, hasta investigaciones de cómo puede afectar el 5G o IPv6 a los ataques DDoS, pasando por comprobar la potencia de un mayor número de dispositivos IoT o investigar la posibilidad de securizar estos dispositivos mediante blockchain.

11.2.1. Investigación para inferir ataques futuros

Otra posible área de investigación sería estudiar la posibilidad de inferir futuros ataques. En este informe se intentaron localizar datos sobre ataques DDoS anuales para realizar una regresión para inferir el número de ataques futuros. Aunque no fuera posible realizarlo en este proyecto, el sistema desarrollado (TrafficReader) podría servir en un futuro para realizar esta investigación.

Para llevar a cabo esta investigación se recomienda instalar el programa TrafficReader o un programa que realice la misma función en los servidores de varias empresas de diferentes sectores. Pasado un tiempo de recogida de datos, que se recomienda que sea al menos un año, se podrían analizar los datos recopilados y estudiar la evolución de los ataques y el tráfico de datos acontecido durante ese tiempo. Se recomienda además que se modifique la base de datos utilizada para considerar el sector de las empresas y otros parámetros que se considerasen útiles para la realización de esta investigación.

Debido a la necesidad de leer el tráfico de red de varias empresas esta investigación debería de realizarla una compañía en el sector de las comunicaciones que contase con los medios necesarios para la recogida y análisis de datos manteniendo estándares de protección y privacidad. Esto facilitaría a dicha empresa ofrecer servicios anti-DDoS adaptados a sus clientes y mejorar su capacidad de ofrecer otros servicios relacionados, aumentando así su presencia en el mercado, mientras que a los clientes que permitieran hacer uso de sus datos para este análisis se les podría hacer una rebaja en el precio del servicio anti-DDoS.

11.2.2. Pruebas con más dispositivos IoT

Actualmente existen multitud de dispositivos IoT diferentes, cámaras, aspiradores, televisores, etc. En este proyecto se han realizado pruebas con dos de ellos, una posible vía de investigación y profundización es la realización de estas pruebas con otros dispositivos, no solo con el fin de investigar su potencia como se ha realizado en este

documento, sino también con la posibilidad de leer las trazas de los paquetes de red enviados que en un futuro podrían servir para ayudar a los servicios anti-DDoS.

11.2.3. 5G e IPv6 en el DDoS

Además del IoT actualmente hay otras tecnologías en desarrollo que podrían afectar de forma significativa a los ataques DDoS. Una de estas tecnologías es el 5G que proporciona más velocidad de conexión a los dispositivos que incorpora.

Otra tecnología se ha desarrollado, pero todavía no se ha implantado, es el protocolo IPv6 que permite tener más IPs diferentes frente al protocolo IPv4. Esto también afecta al control de redes y por lo tanto a los ataques DDoS y servicios anti-DDoS.

Investigar sobre alguna de las dos tecnologías mencionadas en relación con los ataques DDoS sería una línea muy interesante para continuar con este proyecto y considerar la posible evolución de este tipo de ataques.

11.2.4. Blockchain para securizar IoT

Por último, es recomendable investigar sobre tecnologías que puedan utilizarse para securizar los dispositivos IoT. El "Blockchain" es una tecnología conocida debido al "boom" de las criptomonedas, pero que posee el potencial de utilizarse también para securizar este tipo de dispositivos. Durante esta investigación se ha encontrado un estudio sobre el uso de la tecnología blockchain para impedir que algunos dispositivos formen parte de una botnet (Sweeny, 2017).

Profundizar en esta dirección también podría ser otra vía de investigar en la securización de dispositivos IoT y evitar que puedan ser parte de ataques DDoS.

11.3. Recomendaciones

Finalmente, se proponen recomendaciones a empresas, usuarios y gobiernos para reducir la amenaza que suponen los ataques DDoS.

11.3.1. Empresas

Las empresas se benefician de reducir los gastos incurridos. Por ello no todas las empresas cuentan con las medidas de protección necesarias para lidiar contra ataques informáticos. Tras realizar esta investigación y comprobar los altos costes que supone sufrir un ataque DDoS comparado con el gasto que supone contratar servicios anti-DDoS, se recomienda a toda empresa que utilice sistemas informáticos contactar con proveedores de servicios de ciberseguridad para consultar los precios de los servicios anti-DDoS y contratar uno que se adapte a sus necesidades, tamaño y presupuesto.

11.3.2. Particulares

Para los particulares no es necesario contratar servicios de anti-DDoS dado que no es habitual que sufran uno de estos ataques. Sin embargo, sí que pueden tomar medidas para aumentar su seguridad y ayudar a fomentar la seguridad de las empresas que puedan sufrir estos ataques. Para ello, se anima a los usuarios a que adquieran sus productos de aquellos proveedores que se comprometan a proporcionar actualizaciones y resolver posibles fallos de seguridad que pudieran aparecer en dichos dispositivos en el futuro.

11.3.3. Gobiernos

Se proponen dos medidas que los gobiernos pueden tomar para reducir su vulnerabilidad ante ataques DDoS y aumentar la seguridad de los dispositivos IoT.

En primer lugar, se propone que los gobiernos sigan la misma recomendación que se les ha realizado a las empresas, es decir, acudir a proveedores de servicios de ciberseguridad y contratar servicios anti-DDoS que se adapten a sus necesidades, dado que, aun teniendo sus propios departamentos de seguridad informática, el contratar a proveedores externos es beneficioso al proporcionar un segundo nivel de seguridad.

En segundo lugar, se recomienda a los gobiernos promover investigaciones enfocadas a securizar los dispositivos IoT e instar a las empresas proveedoras de estos dispositivos a securizarlos para reducir al máximo posible las infecciones por malware y su pertenencia a botnets, reduciendo así su uso en ataques DDoS.

REFERENCIAS

Andreu_rius. (septiembre, 2017). Weather Station using ESP8266 and some recycled sensors [Pregunta en foro]. Recuperado de https://community.thinger.io/t/weather-station-using-esp8266-and-some-recycled-sensors/615

Anscombe, T. (6 mayo, 2019). Legislar la seguridad de los dispositivos IoT: ¿es realmente la solución? [Post en blog]. Recuperado de https://www.welivesecurity.com/la-es/2019/05/06/legislar-seguridad-dispositivos-iot/

Cáceres, J. (22 octubre, 2016). ¿Cómo ocurrió el ataque DDoS a DynDNS? Recuperado de https://www.apañados.es/tenemos-que-apanar-noticias-internet/hacker/71234-como-ocurrio-el-ataque-ddos-a-dyndns.html

Cloudflare. (2019). UDP Flood Attack [Artículo en web]. Recuperado de https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/

Cox, J. (2 octubre, 2014). La historia de los ataques DDoS como forma de protesta [Post en blog]. Recuperado de https://www.vice.com/es/article/d7dgja/la-historia-de-los-ataques-ddos-como-forma-de-protesta

Del Valle Hernández, L. (2018). NodeMCU tutorial paso a paso desde cero [Post en blog]. Recuperado de https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/

Domodesk. (julio, 2014). A FONDO: ¿QUÉ ES IOT (EL INTERNET DE LAS COSAS)? [Post en web]. Recuperado de https://www.domodesk.com/221-a-fondo-que-es-iot-el-internet-de-las-cosas.html

Drupal, (8 octubre, 2008). Uses of botnets [Post en blog]. Recuperado de https://www.honeynet.org/node/52

EDUCBA. (2019). C vs Python. Recuperado de https://www.educba.com/c-vs-python/

ENISA. (3 noviembre 2016). Major DDoS Attacks Involving IoT Devices [Post en blog]. Recuperado de https://www.enisa.europa.eu/publications/info-notes/major-ddos-attacks-involving-iot-devices

Espinosa, C. (23 mayo, 2017). ¿El primer ataque DoS de la historia? [Artículo en web] Recuperado de https://securityinside.info/el-primer-ataque-dos-de-la-historia/

Fallahi, F. (12 junio, 2014). synflood.py [Código]. Recuperado de https://github.com/fffaraz/Etcetera/blob/master/python/teh1337/synflood.py

Fisher, D. (25 abril, 2013). ¿Qué es un botnet? [Post en blog]. Recuperado de https://www.kaspersky.es/blog/que-es-un-botnet/755/

García Muelas C. (2018). Integración de Redes Telemáticas IoT con Raspberry pi (Trabajo de fin de carrera, Universitat Oberta de Catalunya). Recuperado de http://openaccess.uoc.edu/webapps/o2/bitstream/10609/40187/6/cgmuelasTFC0115memoria.pdf

Goebelbecker, E. (13 diciembre, 2018). Python vs Java: Which is best). Recuperado de https://raygun.com/blog/java-vs-python/

González, G. (15 marzo, 2018). GitHub acaba de sobrevivir el ataque DDoS más grande de la historia [Noticia en web]. Recuperado de https://www.genbeta.com/actualidad/github-acaba-de-sobrevivir-el-ataque-ddos-mas-grande-de-la-historia

Grokhotkov I. (2017a). UDP [Documentación en web]. Recuperado de https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/udp-examples.html

Grokhotkov I. (2017b). Client [Documentación en web]. Recuperado de https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/client-examples.html?highlight=protocol%3A%20tcp

Harán J.M. (7 enero, 2019). 10 principales fallos de seguridad de los dispositivos IoT [Post en blog]. Recuperado de https://www.welivesecurity.com/la-es/2019/01/07/principales-fallos-seguridad-dispositivos-iot/

Ha3MrX. (27 septiembre, 2018). DDos-Attack [Código]. Recuperado de https://github.com/Ha3MrX/DDos-Attack

Hengst K. (2016). DDoS throught the Internet of Things [Paper]. Recuperado de https://www.semanticscholar.org/paper/DDoS-through-the-Internet-of-Things-Hengst/2ab62f22832c96e1f1ed0a43b0a43ed99cc5fcc5#paper-header

Ibragimov, T. Kupreev, O. Badovskaya, E. y Gutnikov, A. (24 julio, 2018). Los ataques DDoS en el segundo trimestre de 2018 [Post en web]. Recuperado de https://securelist.lat/ddos-report-in-q2-2018/87217/

Imperva. (2019a). Ping flood (ICMP flood) [Artículo en web]. Recuperado de https://www.imperva.com/learn/application-security/ping-icmp-flood/

Imperva. (2019b). High Orbit Ion Cannon (HOIC). Recuperado de https://www.imperva.com/learn/application-security/high-orbit-ion-cannon/

Instituto Nacional de Ciberseguridad [INCIBE]. (25 abril, 2019). La importancia de la seguridad en IoT. Principales amenazas [Post en blog]. Recuperado de https://www.incibe-cert.es/blog/importancia-seguridad-iot-principales-amenazas

Institute of Electrical and Electronics Engineers. (1998). IEEE Recommended Practice for Software Requirements Specifications (IEEE830). Recuperado de http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf

Intereconomía. (2 abril, 2018). Los ataques DDoS superan los 2 millones de euros de coste a las empresas [Articulo en web]. Recuperado de https://intereconomia.com/tecnologia/los-ataques-ddos-supera-los-2-millones-de-euros-de-coste-las-empresas-20180402-1046/

it-forensics. (5 agosto, 2014). ping-of-death.py [Código]. Recuperado de https://github.com/it-forensics/forensics/blob/master/src/ping-of-death.py

Jesus, Y. (25 febrero, 2010). SYN FLOOD, QUE ES Y COMO MITIGARLO [Post en blog]. Recuperado de http://www.securitybydefault.com/2010/02/syn-flood-que-es-y-como-mitigarlo.html

Kaspersky. (2014). Global IT Security Risks Survey 2014- Distributed Denia lof Service (DDoS) Attacks [Informe]. Recuperado de https://media.kaspersky.com/en/B2B-International-2014-Survey-DDoS-Summary-Report.pdf

Kaspersky. (2015). Global IT Security Risks Survey [Informe]. Recuperado de http://go.kaspersky.com/rs/802-IJN-240/images/Global%20IT%20Security%20Risks%20Survey%20Ent.pdf

Kaspersky. (2 octubre, 2017). No hay víctimas pequeñas para los cibercriminales [Post en web]. Recuperado de https://www.kaspersky.es/about/press-releases/2017_no-small-victims-for-cybercriminals

Kaspersky, E. (6 diciembre, 2016). Una breve historia de loa ataques DDoS [Post en blog]. Recuperado de https://eugene.kaspersky.com.mx/2016/12/06/una-breve-historia-de-los-ataques-ddos/

Khalimonenko, A. Kupreev, O. y Badovskaya, E. (26 abril, 2018). Los ataques DDoS en el primer trimestre de 2018 [Post en web]. Recuperado de https://securelist.lat/ddos-report-in-q1-2018/86887/

Khalimonenko, A. Kupreev, O. & Ilganaev, K. (6 noviembre, 2017). Los ataques DDoS en el tercer trimestre de 2017 [Post en web]. Recuperado de https://securelist.lat/ddos-attacks-in-q3-2017/85669/

Khalimonenko, A. Kupreev, O. & Ilganaev, K. (6 febrero, 2018b). Los ataques DDoS en el cuarto trimestre de 2017 [Post en web]. Recuperado de https://securelist.lat/ddos-attacks-in-q4-2017/85956/

Kupreev, O. Badovskaya, E. y Gutnikov, A. (31 octubre, 2018). Los ataques DDoS en el tercer trimestre de 2018 [Post en web]. Recuperado de https://securelist.lat/ddos-report-in-q3-2018/88036/

Logrhythm. (26 de julio, 2016). Detecting Beaconing Malware with Network Monitor [Post en blog]. https://es.logrhythm.com/blog/catching-beaconing-malware/

MariaDB. (2019). Documentation. Recuperado de https://mariadb.org/

Martínez, C. (17 marzo, 2017). Ya se han vendido 12.5 Millones de Raspberry Pi [Articulo en web]. Recuperado de https://www.engadget.com/es/2017/03/17/vendido-12-5-millones-raspberry-pi/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAACTyjrNpjRg6zrwl4spknqwnNrY2ZISrd2Vrcms4rVzHhsTIscqb1ltNy3xzTKXFj0tHm7hBHxLAxcCUk23BCSsljrOzgX67PqgnoaoYHZZWL9_RMp_2kb2oIdLX4hJJ0CsQawlDZ_C90xo9PS-9o3HUbG55rHKrOoD45QstcIgx

Masadelante.com. (2019). ¿Qué es un socket? – Definición de Socket [Articulo en web]. Recuperado de https://www.masadelante.com/faqs/socket

MongoDB. (2019). Documentation. Recuperado de https://www.mongodb.com/

Moore, D. Voelker, M. Savage, S. (2001). Inferring Internet Denial-of-Service Activity [Informe]. Recuperado de https://www.caida.org/publications/papers/2001/BackScatter/

Motos, V. (13 diciembre, 2010). LOIC: La herramienta utilizada por Anonymous [Post en blog]. Recuperado de https://www.hackplayers.com/2010/12/loic-la-herramienta-ddos-utilizada-por.html

MySQL. (2019). Documentation. Recuperado de https://www.mysql.com/

Netscout, Arbor Networks. (2018). NETSCOUT Arbor's 13th Annual Worldwide Infrastructure Security Report [Informe en web]. Recuperado de https://pages.arbornetworks.com/rs/082-KNA-087/images/13th_Worldwide_Infrastructure_Security_Report.pdf#WISR_Interactive_FINAL.indd%3A.31785%3A164

Netscout. (2019). NETSCOUT's 14th Annual Worldwide Infrastructure Security Report [Informe en web]. Recuperado de https://www.netscout.com/sites/default/files/2019-03/SECR_005_EN-1901%E2%80%93WISR.pdf

Neustar. (17 enero, 2019a). The Changing Face of Cyber Attacks [Informe]. Recuperado de https://ns-cdn.neustar.biz/creative_services/biz/neustar/www/resources/whitepapers/it-security/neustar-changing-face-of-cyber-attacks-2018.pdf?_ga=2.143628157.955615846.1551896526-770456636.1551896526

Neustar. (17 abril, 2019b). Q1, 2019 Cyber Threats & Trends Report [Informe]. Recuperado de https://ns-cdn.neustar.biz/creative_services/biz/neustar/www/resources/whitepapers/it-security/neustar-cyber-threats-and-trends-report-q1-2019.pdf?_ga=2.48025191.1704921154.1566636814-113225961.1566636814

Packet Sender. (2019). Documentation. Recuperado de https://packetsender.com/

PaloAlto Networks. (2019). WHAT IS A DENIAL OF SERVICE ATTACK (DDoS) [Post en web]. Recuperado de https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos

Pandorafms. (18 noviembre, 2015). Tipos de Bases de datos y las mejores bases de datos [Post en blog]. Recuperado de https://pandorafms.com/blog/es/tipos-de-bases-de-datos-y-las-mejores-bases-de-datos-del-2016/

Pandorafms. (18 julio, 2019). NoSQL vs SQL; principales diferencias y cuando elegir cada una de ellas [Post en web] Recuperado de https://pandorafms.com/blog/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/

Pandya, G. (2 noviembre, 2015). Preparing to withstand a DDoS Attack [Sans Institute Paper]. Recuperado de https://www.sans.org/reading-room/whitepapers/incident/paper/36412

Perakovic, D., Perisa, M., Cvitic, I. (2 diciembre 2015). ANALYSIS OF THE IoT IMPACT ON VOLUME OF DDoS ATTACKS [Paper]. Recuperado de

http://postel.sf.bg.ac.rs/simpozijumi/POSTEL2015/RADOVI%20PDF/Telekomunikacioni%20servisi%20-%20kvalitet%20i%20ekonomski%20aspekti/5.%20Perakovic-Perisa-Cvitic.pdf

Ponemon Institute (noviembre, 2012). Cyber Security on the Offense:A Study of IT Security Experts [Informe]. Recuperado de https://security.radware.com/uploadedFiles/Resources_and_Content/Attack_Tools/CyberSecurityontheOffense.pdf

Professor Messer, (7 septiembre, 2014). Christmas Tree Attack – CompTIA Security+ SY0-401: 3.2 [Post en blog]. Recuperado de https://www.professormesser.com/security-plus/sy0-401/christmas-tree-attack-2/

Puri, R., (8 agosto, 2003). Bots &; Botnet: An Overview [Sans Institute Paper]. Recuperado de https://www.sans.org/reading-room/whitepapers/malicious/bots-botnet-overview-1299

Python. (2019). Documentation. Recuperado de https://www.python.org/

Python. (26 Mayo, 2019). socket — Low-level networking interface [Documentacion]. Recuperado de https://docs.python.org/3/library/socket.html#socket.AF_INET.

Qlik. (2019). Documentation. Recuperado de https://www.qlik.com/es-es

Raspberry. (2019). Raspberry Information. Recuperado de https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Rouse M. (enero, 2017). Internet de las cosas (IoT) [Post en blog]. Recuperado de https://searchdatacenter.techtarget.com/es/definicion/Internet-de-las-cosas-IoT

Silver Moon. (29 noviembre, 2011). Code a network packet sniffer in python for Linux [Código en el artículo]. Recuperado de https://www.binarytides.com/python-packet-sniffer-code-linux/.

Sweeny, J. (7 septiembre, 2017). Botnet Resiliency via Private Blockchains. [Paper]. Recuperado de https://www.sans.org/reading-room/whitepapers/covert/botnet-resiliency-private-blockchains-38050

Systemadmin, (27 agosto, 2009). slowloris: Ataque de denegación de servicio para apache 1.x y 2.x [Post en blog]. Recuperado de http://systemadmin.es/2009/08/slowloris-ataque-de-denegacion-de-servicio-para-apache-1x-y-2x

The Cyber Security Hub (9 junio, 2019). [Video]. Recuperado de https://www.linkedin.com/feed/update/urn:li:activity:6543189371565285376/

Thinguer.io. (2019). Documentation. Recuperado de https://thinger.io/

ACRÓNIMOS

UDP – User Datagram Protocol

TCP – Transmission Control Protocol

ICMP – Internet Control Message Protocol

IoT – Internet of Things

DoS – Denial of service

DDoS – Distributed Denial of service

DNS – Domain Name System

IP – Internet Protocol

INCIBE – Instituto Nacional de Ciberseguridad

ENISA – Agencia de la Unión Europea para la ciberseguridad

ANEXO 1: PLANIFICACIÓN DEL TRABAJO

En esta sección se detalla la planificación inicial para el desarrollo del trabajo, así como el tiempo que realmente supuso desarrollar cada fase del proyecto y se señalan las desviaciones entre el plan inicial y el progreso real del proyecto.

1. Planificación inicial

En la planificación inicial se estimó que se iban a tardar unos 144 días en realizar el proyecto, de los cuales los primeros 28 días serían dedicados a la investigación, tanto de los ataques DDoS como de los dispositivos IoT, y finalmente de la relación de estos conceptos con las botnets. Los 44 días posteriores serían dedicados al desarrollo del código, incluyendo la investigación necesaria para encontrar la mejor manera de desarrollar el programa, el desarrollo en sí y las gestiones y pruebas necesarias para evaluar su funcionamiento.

Una vez desarrollado el código, el plan inicial especificaba 22 días para realizar las pruebas para comprobar la capacidad de un dispositivo IoT de realizar un ataque DDoS y analizar sus datos. Finalmente, otros 29 días fueron planificados para investigar sobre los costes que sufrir ataques DDoS causaba a las empresas y 21 días se reservaron para la redacción y edición del documento.

La tabla 20 muestra los tiempos planificados para la realización de cada tarea, las fechas de inicio y fin programadas y las fases que necesitaban ser completadas previamente para poder realizar cada una de las tareas (dependencias).

Tabla 20

Planificación de tareas inicial.

ID	Tarea	Tiempo	Fecha Inicio	Fecha Fin	Dependencias
	TFG	144d	01/02/19	24/06/19	
	Investigación	28d	01/02/19	28/02/19	
1	Investigación ataques DDoS	7d	01/02/19	07/02/19	
2	Investigación IoT	7d	08/02/19	14/02/19	
3	Investigación IoT y DDoS y Botnets	14d	15/02/19	28/02/19	1; 2
	Código	44d	01/03/19	13/04/19	
4	Investigación sobre códigos para leer tráfico entrante y plantear sistemas a utilizar	15d	01/03/19	15/03/19	3
5	Desarrollar el código	17d	16/03/19	01/04/19	4
6	Levantar servidor	5d	02/04/19	06/04/19	5
7	Semana de control del servidor	7d	07/04/19	13/04/19	6
	Pruebas	22d	14/04/19	05/05/19	
8	Realización de las pruebas	9d	14/04/19	22/04/19	7
9	Análisis de los datos obtenidos	13d	23/04/19	05/05/19	8
	Investigación Costes DDoS	29d	06/05/19	03/06/19	
10	Localizar Informes	7d	06/05/19	12/05/19	2
11	Analizar Informes	9d	13/05/19	21/05/19	10
12	Comparar informes	13d	22/05/19	03/06/19	11
13	Creación del documento	21d	04/06/19	24/06/19	6; 12

Nota. Fuente: Elaboración propia

2. Diagrama de Gantt

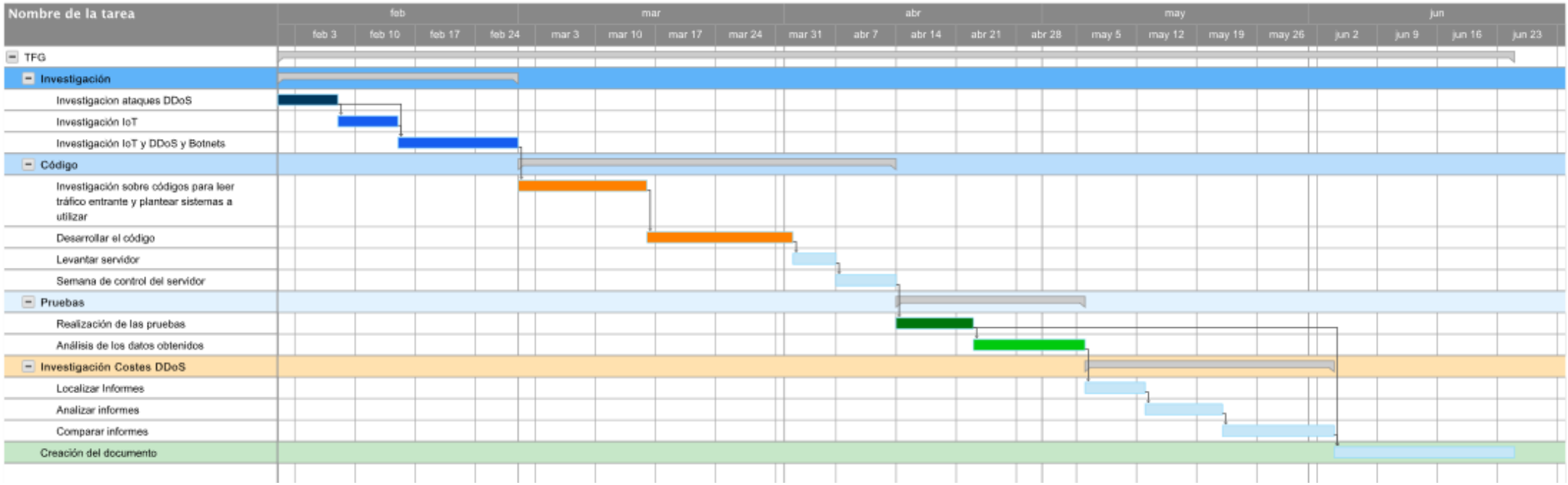

Figura 39 Diagrama de Gantt inicial. Fuente: Elaboración propia.

3. Tiempo real y desvío de horas

Finalmente, el trabajo llevó 202 días, 58 días más de los que fueron planificados para su realización. Esta desviación fue mayormente debida a la dificultad de localizar informes y bases de datos sobre los ataques DDoS y sus costes. La primera investigación sobre ataques DDoS, IoT y su relación con las botnets llevó 34 días, que corresponde a un 21.4% más respecto a la cantidad de días inicialmente planificada. Se tardaron 52 días en desarrollar el código, lo que supuso una demora de 8 días respecto al plan inicial. La realización de las pruebas se llevó a cabo en 27 días, un tiempo también ligeramente (5 días) superior al planeado. Finalmente, la Investigación sobre los costes DDoS y la compilación del trabajo llevaron 49 y 40 días respectivamente, suponiendo un aumento del 69% y del 90% respectivamente respecto a los tiempos inicialmente especificados para estas tareas (22 días y 29 días).

La tabla 21 muestra el tiempo real de la realización de la investigación y su desvío frente a lo estimado. En aquellas tareas en las que el desvío es positivo, y por lo tanto han llevado más tiempo que el planeado, se ha usado una fuente de color rojo. En aquellas tareas en las que el desvío es negativo, es decir, que han llevado menos tiempo que el planeado, se ha usado una fuente de color verde.

Tabla 21

Planificación de las tareas real.

Tarea	Tiempo	Fecha Inicio	Fecha Fin	Desvío Frente al inicial	Dependencias
TFG	202d	01/02/19	01/09/19	+40,2%	
Investigación	34d	01/02/19	06/03/19	+21,4%	
Investigación ataques DDoS	10d	01/02/19	10/02/19	+42,8%	
Investigación IoT	10d	11/02/19	20/02/19	+42,8%	
Investigación IoT y DDoS y Botnets	14d	21/02/19	06/03/19	0	1; 2
Código	52d	07/03/19	27/04/19	+18,2%	
Investigación sobre códigos para leer tráfico entrante y plantear sistemas a utilizar	20d	07/03/19	26/03/19	+33,3%	3
Desarrollar el código	22d	27/03/19	15/04/19	+17,6%	4
Levantar servidor	3d	16/04/19	20/04/19	- 40%	5
Semana de control del servidor	7d	21/04/19	27/04/19	0	6
Pruebas	27d	28/04/19	24/05/19	+22,7%	
Realización de las pruebas	12d	28/04/19	09/05/19	+33,3%	7
Análisis de los datos obtenidos	15d	10/05/19	24/05/19	+15,4%	8
Investigación Costes DDoS	49d	25/05/19	12/07/19	+69%	
Localizar Informes	17d	25/05/19	10/06/19	+142,8%	2
Analizar informes	13d	11/06/19	23/06/19	+44,4%	10
Comparar informes	19d	24/06/19	12/07/19	+46,1%	11
Creación del documento	40d	13/07/19	01/09/19	+90%	6; 12

Nota. Fuente: Elaboración propia

4. Diagrama de Gantt actualizado

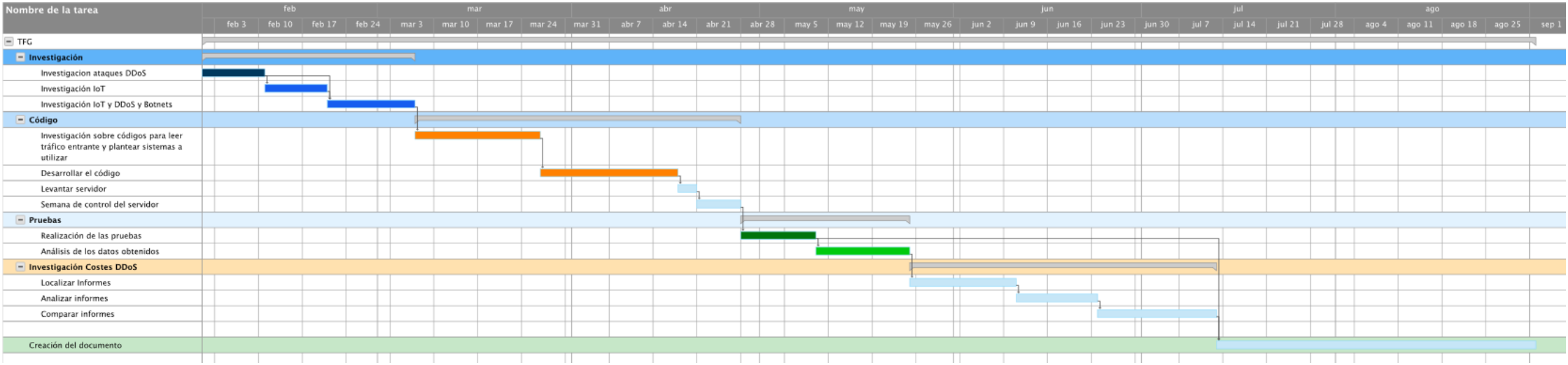

Figura 40 Diagrama de Gantt actualizado. Fuente: Elaboración propia

ANEXO 2: COSTE DE LA INVESTIGACIÓN

1. Coste de la mano de obra

El trabajo se ha planificado llevarlo a cabo en 144 días, incluyendo fines de semana. Se ha planificado un trabajo de cinco horas al día. Esto supone una inversión total para la realización del trabajo de 720 horas lo que, asumiendo un salario de 15,50€ la hora, conlleva un gasto total de 11.162€. Esta cifra incluye los impuestos, tanto a cargo de la empresa como a cargo del trabajador. La tabla 22 muestra el reparto de este coste.

Tabla 22

Coste de la mano de obra.

Sueldo Neto del trabajador	Impuestos
6987,41€	4174,59€

Nota. Fuente: Elaboración propia

2. Elementos

La tabla 23 muestra los elementos utilizados, su precio de venta (P.V.P.), su vida útil estimada y el coste imputado al proyecto. Este coste ha sido estimado utilizando una depreciación lineal sin valor residual y considerando los días de uso de los elementos los 144 días que está planificado.

Tabla 23

Coste de los elementos utilizados.

Elementos	P.V.P.	Tiempo de vida Útil	Coste imputado
Raspberry pi 3	35,75€	3 años	4,70€
Node MCU	1,86€	1 año	0,73€
HP-Omen 15-ce0XX	1100€	6 años	72,33€
Toshiba Satellite C650	100€	10 años	3,95€
Coste Total Imputado			81,71€

Nota. Fuente: Elaboración propia

3. Otros gastos

Para el cómputo de los gastos de este proyecto también se tienen que considerar otros costes directos (CD) como la impresión de la documentación, y costes indirectos (CI) como la luz y el internet utilizados. No se ha contabilizado el coste de los softwares

utilizados debido a que se han utilizado solamente versiones gratuitas de los mismos o programas cuya licencia es proporcionada por la Universidad Carlos III de Madrid.

La tabla 24 muestra estos otros costes. Para valorar los costes imputados por los gastos indirectos, se ha multiplicado su coste total en el tiempo que ha durado el proyecto y multiplicado por 4/24, número que representa la fracción diaria en la que los sistemas eran utilizados y se han redondeado las decenas hacia abajo ($CI=\frac{CosteMensual}{30}*144*\frac{5\ Horas}{24\ Horas}$).

Tabla 24

Otros costes.

Concepto	Coste Mensual	Coste imputado
Internet	50€	50€
Luz	55€	50€
Fotocopias	50€	50€
Coste Total Imputado		150€

Nota. Fuente: Elaboración propia

4. Coste total del proyecto

El coste total del proyecto sería la suma de los tres totales anteriores. El proyecto finalmente ha costado un **total de 11.393,71€,** siendo la mano de obra el mayor gasto asumido. La tabla 25 muestra el concepto y los costes imputados al proyecto.

Tabla 25

Coste Total del proyecto.

Concepto	Coste imputado
Gastos de personal	11.162€
Raspberry pi 3	4,70€
Node MCU	0,73€
HP-Omen 15-ce0XX	72,33€
Toshiba Satellite C650	3,95€
Internet	50€
Luz	50€
Fotocopias	50€
Coste Total sin IVA	11.393,71€
IVA (21%)	2392,68
Total	13786,39

Nota. Fuente: Elaboración propia

5. Presupuesto para cliente

En esta sección se expone el presupuesto a presentar al cliente. En él se agrupan los gastos, el beneficio y el riesgo que permita hacer frente a los imprevistos que surjan.

El porcentaje de riesgo del proyecto se ha definido a 10%. Este valor se ha definido en concordancia con los valores utilizados en anteriores proyectos de la carrera.

Se ha marcado que se añadirá un 35% al coste del proyecto en concepto de beneficios. Este beneficio se ha marcado de acuerdo a proyectos realizados anteriormente. Estos beneficios serían los únicos obtenidos directamente por esta investigación. Posteriormente esta investigación podría ser utilizada para ayudar a la hora de vender otros productos como el anti-DDoS, pero no obtendría ingresos por si sola.

La tabla 26 muestra el presupuesto a entregar al cliente en relación con esta investigación.

Tabla 26

Presupuesto.

Concepto	Coste imputado
Gastos de personal	11.162€
Gastos de Hardware	81,71€
Gastos directos	50€
Gastos indirectos	100€
Total Gastos sin riesgo	**11.393,71€**
Riesgo (10%)	1.139,37€
Total Gastos sin beneficios	**12.533,08€**
Beneficios (35%)	4.386,57€
Total Gastos sin IVA	**16.919,66**
IVA (21%)	3.553,13
Total	**20.472,79**

Nota. Fuente: Elaboración propia

6. Coste final y desviación

En este apartado se analiza el coste final en el que se ha incurrido y su desviación frente al presupuestado.

La tabla 27 muestra los costes presupuestados en cada uno de los apartados, el coste real y la variación entre los mismos, así como el porcentaje de variación que corresponde.

Tabla 27

Diferencia entre presupuesto y real.

Concepto	Coste Presupuestado	Coste Real	Variación	Porcentaje
Gastos de personal	11.162€	12.524€	1362€	12,2%
Gastos de Hardware	81,71€	134.89€	53,18€	65,08%
Gastos directos	50€	50€	0€	0%
Gastos indirectos	100€	90€	-10€	-10%
Total	**11.393,71€**	**12.798,89€**	**1405,18€**	**12,33%**

Nota. Fuente: Elaboración propia

Se ha incurrido en un coste 1405,18€ superior al presupuestado en un inicio, lo que equivale a un 12,33% del presupuestado. El coste al cliente seguirá siendo el mismo. Esto se debe a que se reservó un 10% del total presupuestado para riesgos, lo que equivale a 1139,37€. Restando esa cantidad al total gastado queda un total de 265,81€ de gastos por encima de lo presupuestado. Esos gastos se retirarán del beneficio presupuestado y se obtiene un beneficio total de 4.120,76€

La variación en el coste respecto al presupuestado se debe en su mayor parte al incremento en días de la duración del desarrollo del proyecto. En vez de durar 144 días se terminó llevando a cabo en 202. Por otra parte, se estipuló que se trabajarían 5 horas diarias cuando finalmente fueron cuatro horas diarias, eso provoca que la desviación en el coste del proyecto no sea tan elevada y que prácticamente el porcentaje de riesgo se haga cargo de la mayor parte de la desviación.

Uno de los mayores errores que se cometió en la planificación fue plantear el trabajo diario superior al tiempo que se disponía para su realización, esto se irá afinando en proyectos posteriores.

ANEXO 3: CÓDIGOS UTILIZADOS

La siguiente tabla 28 contiene 3 enlaces a los 3 códigos utilizados para el desarrollo del *TrafficReader* los códigos serán publicados el día 24 de septiembre de 2019.

Tabla 28

Códigos utilizados.

Código	Enlace
Lector TCP	https://github.com/trilogi77/TFG/tree/master/TCP
Lector UDP	https://github.com/trilogi77/TFG/tree/master/UDP
Lector ICMP	https://github.com/trilogi77/TFG/tree/master/ICMP
Código Ataque TCP NodeMCU	https://github.com/trilogi77/TFG/tree/master/AtaqueTCPNodeMCU
Código Ataque UDP NodeMCU	https://github.com/trilogi77/TFG/tree/master/AtaqueUDPNodeMCU

Nota. Fuente: Elaboración propia

ANEXO 4: PLANTILLAS

1. Plantilla para la especificación de requisitos software

Tabla 29

Plantilla para la especificación de requisitos de software.

Requisitos del software				
Tipo: Funcional /No Funcional				
Id	Nombre	Descripción	Estabilidad	Prioridad
RF-XX/ RNF-XX			Alta/Media/Baja	Alta/Media/Baja

Nota. Fuente: Elaboración propia

2. Plantilla para la definición de casos de uso

Tabla 30

Plantilla para la definición de casos de uso.

Caso de uso	Nombre
ID	CU-XX
Actores	
Descripción	
Precondiciones	
Postcondiciones	

Nota. Fuente: Elaboración propia

3. Plantilla para la especificación de las pruebas de aceptación

Tabla 31

Plantilla para la especificación de las pruebas de aceptación.

ID prueba	PA-XX
Descripción	
Requisitos a comprobar	
Precondición	
Acción	
Postcondición	

Nota. Fuente: Elaboración propia